March 2020

Learning Latent Characteristics of Data and Models using Item Response Theory

John P. Lalor
University of Notre Dame

LEARNING LATENT CHARACTERISTICS OF DATA AND MODELS USING ITEM RESPONSE THEORY

A Dissertation Presented

by

JOHN P. LALOR

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

February 2020

College of Information and Computer Sciences

LEARNING LATENT CHARACTERISTICS OF DATA AND MODELS USING ITEM RESPONSE THEORY

A Dissertation Presented

by

JOHN P. LALOR

Approved as to style and content by:

Hong Yu, Chair

James Allan, Member

Brendan O'Connor, Member

Lisa Keller, Member

James Allan, Chair of the Faculty
College of Information and Computer Sciences

DEDICATION

To my family.

Ad maiorem Dei gloriam

St. Ignatius of Loyola

I bpoll sa talamh a bhí cónaí ar hobaid

J.R.R. Tolkien

ACKNOWLEDGMENTS

This is the part where I thank everyone who made this possible.

I have to start with my advisor, Hong Yu. Throughout my time at UMass, you have supported me, encouraged me, and, most importantly, challenged me. I've learned so much from you during my time at UMass. To my dissertation committee, James Allan, Brendan O'Connor, and Lisa Keller, thank you for the very helpful input and suggestions throughout this process. Your help has only improved what is presented here. Hao Wu, Kathy Mazor, and Bev Woolf have been wonderful collaborators for several research projects. Their guidance and suggestions have helped strengthen all of the work presented here. The members of the BioNLP lab have been sounding boards, cheerleaders, and friends, among other things. Abhyuday, Jesse, Jiaping, Subhendu, Bhanu, Frank, Alice, Tsendee, Jinying, Emily, Elaine, Weisong, Kathryn, and everyone else, thank you all. Thank you to everyone at CICS, especially Leanne LeClerc and Eileen Hamel.

Before I started at UMass I was a part-time Masters student at DePaul taking a full-time course load. It was there that I first thought of research as a career, and I must thank the great professors at DePaul who worked with me to develop that interest and expose me to several different areas within CS: Amber Settle, Terrie Steinbauch, Craig Miller, Robin Burke, Jonathan Gemmell. To Rob Easley and everyone in the Mendoza College of Business at Notre Dame, that is where it all started for me, and I am so excited to be returning.

During my Ph.D. I had the opportunity to spend summers working at ESPN and Amazon. For a life-long athlete, getting to see how the sausage was made in

Bristol, CT really was a dream come true. To Zvi Topol, Javid Husenov, Adithya Tammavarapu, Sean Sanders, Segun Oshin, and everyone in the ESPN Advanced Technology Group who made that a very enjoyable and productive summer, thank you. Imre Kiss, Bill Campbell, Eunah Cho, Francois Marrissee and the rest of the Alexa team in Cambridge, MA pushed me to take what I'd learned and put it into practice, while also thinking through new research ideas.

When I was young, there was a rule in my house with regards to books: you can have them. Mom and Dad, you have always supported me and my interests. You pushed me when I was young to stay focused and challenge myself, in all aspects of life. Thank you. Kevin, you are and have always been my best friend. I can talk to you about anything, and spending time with you and your wife (!) Meghan on trips home to Philly has always been a great way to unwind and leave the stresses of work behind.

I would not be here if it weren't for the love and support of my wife Kaitlin. From Notre Dame to Chicago to Western Mass. and back to ND, you have been my rock, my support, and my inspiration. During the highs and lows and stresses and more stresses of this process, you have always been there for me. I love you.

I dedicate this thesis to our girls, Teresa and Maeve, and to our upcoming third child. Tess and Maeve, watching you grow and learn has been the greatest experience of my life. I can't wait to see the women that you become. BL3, we're very excited to meet you in March!

ABSTRACT

LEARNING LATENT CHARACTERISTICS OF DATA AND MODELS USING ITEM RESPONSE THEORY

FEBRUARY 2020

JOHN P. LALOR

B.B.A., UNIVERSITY OF NOTRE DAME

M.Sc., DEPAUL UNIVERSITY

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Hong Yu

A supervised machine learning model is trained with a large set of labeled training data, and evaluated on a smaller but still large set of test data. Especially with deep neural networks (DNNs), the complexity of the model requires that an extremely large data set is collected to prevent overfitting. It is often the case that these models do not take into account specific attributes of the training set examples, but instead treat each equally in the process of model training. This is due to the fact that it is difficult to model latent traits of individual examples at the scale of hundreds of thousands or millions of data points. However, there exist a set of psychometric methods that can model attributes of specific examples and can greatly improve model training and evaluation in the supervised learning process.

Item Response Theory (IRT) is a well-studied psychometric methodology for scale construction and evaluation. IRT jointly models human ability and example characteristics such as difficulty based on human response data. We introduce new evaluation

metrics for both humans and machine learning models build using IRT, and propose new methods for applying IRT to machine learning-scale data.

We use IRT to make contributions to the machine learning community in the following areas: (i) new test sets for evaluating machine learning models with respect to a human population, (ii) new insights about how deep-learning models learn by tracking example difficulty and training conditions, and (iii) new methods for data selection and curriculum building to improve model training efficiency, (iv) a new test of electronic health literacy built with questions extracted from de-identified patient Electronic Health Records (EHRs).

We first introduce two new evaluation sets built and validated using IRT. These tests are the first IRT test sets to be applied to natural language processing tasks. Using IRT test sets allows for more comprehensive comparison of NLP models. Second, by modeling the difficulty of test set examples, we identify patterns that emerge when training deep neural network models that are consistent with human learning patterns. Specifically, as models are trained with larger training sets, they learn easy test set examples more quickly than hard examples. Third, we present a method for using soft labels on a subset of training data to improve deep learning model generalization. We show that fine-tuning a trained deep neural network with as little as 0.1% of the training data can improve model generalization in terms of test set accuracy. Fourth, we propose a new method for estimating IRT example and model parameters that allows for learning parameters at a much larger scale than previously available to accommodate the large data sets required for deep learning. This allows for learning IRT models at machine learning scale, with hundreds of thousands of examples and large ensembles of machine learning models. The response patterns of machine learning models can be used to learn IRT example characteristics instead of human response patterns. Fifth, we introduce a dynamic curriculum learning process that estimates model competency during training to adaptively select training data that is appropriate

for learning at the given epoch. Finally, we introduce the ComprehENotes test, the first test of EHR comprehension for humans. The test is an accurate measure for identifying individuals with low EHR note comprehension ability, and validates the effectiveness of previously self-reported patient comprehension evaluations.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

INTRODUCTION

A typical supervised learning setup in machine learning involves using a large annotated training data set to fit a model capable of learning patterns in the training data in such a way that the model can generalize to an unseen test data set. Learning involves updating the model parameters according to differences between the model output and a single true, gold-standard label. The input data, which interacts with the model weights to provide the model output, is often taken as given. The output also is relatively static when compared to the work on model tuning. The gold-standard is the gold-standard, and we want our model to fit well to the data while also being able to generalize well. These gold-standard examples are fixed, and specific characteristics of the examples do not affect evaluation.

Once trained, model performance is evaluated by labeling a previously unseen data set and comparing the output labels to the known, gold-standard labels for that data set. Accuracy, recall, precision and F1 scores are commonly used to evaluate NLP applications. These metrics assume that each point in the data set has equal weight for evaluating performance. However examples are different. Some may be so hard that most/all NLP systems answer incorrectly; others may be so easy that every NLP system answers correctly. Neither example type provides meaningful information about the performance of an NLP system. Examples that are answered incorrectly by some systems and correctly by others are useful for differentiating systems according to their individual characteristics.

We propose an integration of psychometrics and machine learning to better model the supervised learning task. This integration allows for the modeling of input data latent traits as well as model latent traits to both inform the model-training procedure

and provide more insight into model generalization performance. The psychometric methodologies used here are known as Item Response Theory (IRT) [Baker, 2001,Baker and Kim, 2004].

IRT is one of the most widely used methodologies in psychometrics for scale construction and evaluation. It is typically used to analyze human responses (graded as right or wrong) to a set of questions (called "items" in the psychometric literature and examples here). With IRT, individual ability and example characteristics are jointly modeled to predict performance [Baker and Kim, 2004]. This statistical model assumes the following: (a) Individuals differ from each other on an unobserved latent trait dimension (called "ability" or "factor"); (b) The probability of correctly answering an example is a function of the person's ability and of the example's latent parameters. This function is called item characteristic curve (ICC) and involves example characteristics as parameters; (c) Responses to different examples are independent of each other for a given ability level of the person ("local independence assumption"); (d) Responses from different individuals are independent of each other.

First, we introduce two new test sets for natural language inference and sentiment analysis built using IRT that measure the latent ability of a natural language processing model as opposed to raw accuracy, and show that these tests provide more insight into model performance than traditional evaluation such as accuracy or F1. By using IRT, the latent characteristics of specific test set examples affect a model's score. At the same time, the latent ability parameter of a model places the model on a continuum of ability with other test-takers, which allows for comparison between models more informative than a simple accuracy score. With IRT, we show that high accuracy is not necessarily indicative of high performance if a test data set is very easy.

Second, we show that by modeling the difficulty of test set examples, patterns emerge when training deep neural network models that are consistent with human learning patterns, specifically, that as models are trained with larger training sets,

they learn easy test set examples more quickly than hard items. We find that there is a relationship between example difficulty and model performance, not only for fully trained models but also as a function of the training set size used to train the model. This allows for new insights into how models behave under different training circumstances, and quantitatively confirms insights about learning that have been used in methods such as curriculum learning.

Third, we propose a soft-label memorization-generalization training sequence for deep neural networks that leverages human uncertainty about data to fine-tune deep learning models. Soft labels for a small sample of data points are estimated by calculating the distribution over potential labels gathered from Amazon Mechanical Turk workers. By fine-tuning three representative deep learning architectures with soft labels we are able to improve test set performance.

Fourth, we propose a new method for modeling latent example and model characteristics using IRT at a large scale. At present there has not been work done to build very large scale IRT models because the models are typically used to evaluate humans. We use variational inference methods to estimate the latent parameters that allow for much larger scale modeling of the data than previously done.

Fifth, we propose a dynamic data selection strategy for curriculum learning that estimates model competency during training in order to select training data examples that are most appropriate for a learner at a point in time. This allows for selecting training examples based on model competency and not a rigid learning schedule. Dynamic data selection leads to more efficient and effective models.

Finally, we introduce a new test for measuring human Electronic Health Record (EHR) note comprehension and conduct experiments that demonstrate the ability of active educational interventions to improve note comprehension in patients. In the past patient understanding of their EHR notes has only been measured by self-reported patient data. We have developed a test using IRT to evaluate patient latent ability

for EHR note comprehension. This test is the first of its kind, and all questions in the test were automatically identified and extracted from de-identified patient EHR notes. The test demonstrates a real-world use case for the IRT test construction methods in the important area of patient health literacy, specifically with regards to EHRs and EHR notes.

In this dissertation we introduce new methods for test set construction and model evaluation for the machine learning and natural language processing community. In addition, we introduce a new way to learn IRT latent parameters for data sets at machine learning scale that tightly integrates input data information and parameter updating for improved generalization. We demonstrate that the integration of psychometrics into machine learning model training allows for more information about a data set to be used when training a model, leading to more efficient and effective learning. Finally, we present a new test for patient health literacy that will hopefully contribute to future research on measuring and improving patient health literacy. It is our hope that the methods proposed here provide researchers with new methods for training and evaluating models that do more than just use data but take properties of the data and models into account.

CHAPTER 1

BACKGROUND, FOUNDATIONS, AND NOTATION

Our goal is to bring psychometrics to machine learning by demonstrating the usefulness of psychometric methods, specifically Item Response Theory (IRT), on machine learning model training and evaluation. In this chapter we will provide an introduction to IRT for machine learning researchers, and an overview of the machine learning models and training methods that will be used to demonstrate the effectiveness of IRT for the benefit of psychometricians.

In typical machine learning evaluation, aggregate scores such as accuracy are calculated on a held-out test set. The characteristics of individual test set examples such as difficulty are not taken into consideration. Often times, the difficulty of a data set is determined after the fact, once it has been shown that certain baseline models do not do particularly well on the task. There is a need to model the intrinsic difficulty of the data sets used in machine learning to help guide progress in the field and to help place the progress of new models into context. For example, if a new machine learning model outperforms the state-of-the-art for a particular task by 0.01%, what does that really tell us about the new model? It could be that this new model labeled all but 3 test examples the exact same way as the previous state-of-the-art model. But for those 3, if the new model labels the easiest one incorrectly and the two harder examples correctly, while the prior model labeled the easiest example correctly but labeled the the two harder examples incorrectly, what does that mean in terms of which model should be used moving forward? Or, what does that tell us about the data set in question? Even if the new model achieves state-of-the-art performance, is

it acceptable that the model labels the easiest example incorrectly? In order to even know which example is the easiest, there needs to be a way to estimate the difficulty of each example.

Psychometrics is a field in psychology concerned with the evaluation of humans and the design of tests to evaluate those humans. IRT models are psychometric models that estimate the latent ability of humans in certain areas based on their responses to a carefully selected set of examples. These examples also have latent parameters such as difficulty that are learned by gathering a large number of response patterns from individuals. To date, there has been very little work on applying IRT methods in the machine learning community. We propose applying IRT methods to model latent characteristics of supervised learning models and of the data used to train and evaluate them. Specifically, we propose and evaluate the following thesis:

Estimating the characteristics of individual data points such as difficulty and latent model ability using psychometric methods can be done at a large scale, can improve model performance, and can allow for more thorough model evaluation.

1.1 Foundations

The methods described in this thesis apply to supervised machine learning models. For consistency we now define terms that will be used in the subsequent chapters. When there is inconsistency between the IRT and machine learning terminology it will be explicitly mentioned below, and the machine learning terminology will be used moving forward.

Definition 1.1.1 (Example)**.** An example d is a tuple $d = (x, y)$, where x is a set of features associated with the example, and y is the gold-standard label for the example. Each y comes from a set of labels $Y^* = \{y_0, y_1, \ldots, y_{n-1}\}$, where n is the number of possible class labels for the task. Y^* is task-specific. For example, for the task of

sentiment analysis $n = 2$ and $Y^* = \{positive, negative\}$. Examples are referred to as *items* in the IRT literature.

Definition 1.1.2 (Data set)**.** A data set is a collection of examples $D = \{d_0, d_1, \ldots, d_{n-1}\}$, where n is the number of examples in the data set. X^D is the set of features associated with the examples in D, where X_0^D refers to the features of the first example in D. Y^D is the set of gold-standard labels associated with the examples in D, where Y_0^D refers to the gold-standard label of the first example in D. Data sets may be used for model training or evaluation. Data sets used for training are *training sets*. Data sets used for evaluation are *test sets* (typically referred to as *evaluation scales* in the IRT literature).

Definition 1.1.3 (Model)**.** A model provides label predictions for a test set. More formally, for some test set D_{test}, a model M generates label predictions $\hat{Y}^{D_{test}}$ based on the features of D_{test}, $X^{D_{test}}$: $\hat{Y}^{D_{test}} = M(X^{D_{test}})$, which are compared to the gold-standard labels $Y^{D_{test}}$. A model is analogous to a *subject* in the IRT literature. In this work model will refer to a machine learning model, and if humans are involved they will be referred to as subjects.

Definition 1.1.4 (Response Pattern)**.** A response pattern is a binary vector that compares a model's label predictions for a test set with the gold-standard labels. For any model M and data set D, M's response pattern is defined as:

$$Z^{M,D} = [\mathbb{I}[\hat{y}_0 = y_0], \ldots, \mathbb{I}[\hat{y}_n = y_n]] \tag{1.1}$$

where $\mathbb{I}[A]$ is the indicator function, which evaluates to 1 when the expression A is true and 0 when the expression is false.

1.2 Supervised Learning Evaluation

The goal of this thesis is to demonstrate the usefulness of psychometric methods, specifically IRT, for training and evaluating supervised machine learning models. In a typical supervised learning setup, a model is trained on some labeled data set which consists of features and labels. Each example is defined by the features associated with it and each example has a corresponding gold-standard label.

Current gold-standard data set generation methods include web crawling [Guo et al., 2013], automatic and semi-automatic generation [An et al., 2003], and expert [Roller and Stevenson, 2015] and non-expert human annotation [Bowman et al., 2015, Wiebe et al., 1999]. In each case validation is required to ensure that the data collected is appropriate and usable for the required task. Automatically generated data can be refined with visual inspection or post-collection processing. Human annotated data usually involves more than one annotator, so that comparison metrics such as Cohen's or Fleiss' κ can be used to determine how much they agree. Disagreements between annotators are resolved by researcher intervention or by majority vote.

Evaluating these models requires a set of labeled data that was previously unseen by the model, to determine how well the model can generalize outside of the data the model was trained on. This held out test set is typically drawn from the same distribution as the training data. Model evaluation therefore consists of having the trained model generate labels for the test set and comparing these with the gold-standard labels.

There are many methods for how the test sets are obtained. For large data sets, there is typically a pre-defined held out test set to facilitate direct comparison between models. For smaller data sets, methods such as cross-validation are used, where the full data set is split into folds, and copies of the models are trained on all folds but one, which is held out for testing. Evaluation statistics across models are aggregated. We focus on model evaluation via a standard, held-out test set. This is the norm for

evaluating deep learning models on large benchmark data sets, as they are typically released with a pre-defined test set for model comparisons.

To evaluate model training, one typically considers accuracy on the training set. Continual improvement in training set accuracy indicates that the model is "learning" by being better able to classify the instances to which it has been exposed. Most common in machine learning experiments is the arithmetic mean:

Definition 1.2.1 (Training error)**.** The training error of a model M refers to the percentage of examples in a training set D_{train} that the model labels incorrectly:

$$e_{train} = 1 - \frac{1}{N} \sum_{n=1}^{N} z^{M_i, D_{train}} \tag{1.2}$$

Once a model has been trained, generalization performance is measured by the arithmetic mean on the held-out test set.

Definition 1.2.2 (Test error)**.** The test error of a model M refers to the percentage of examples in a test set D_{test} that the model labels incorrectly:

$$e_{test} = 1 - \frac{1}{N} \sum_{n=1}^{N} z^{M_i, D_{test}} \tag{1.3}$$

Other performance metrics exist but are less common in the ML literature. For example, the geometric mean uses the product of responses instead of the sum. This more strictly penalizes incorrect answers.

Definition 1.2.3 (Geometric mean)**.** For some response pattern Z the geometric mean is:

$$(\prod_{i=1}^{N} z_i^M)^{\frac{1}{N}} \tag{1.4}$$

1.3 Item Response Theory

IRT is a methodology of evaluation for characterizing test examples and estimating subject ability from their performance on such tests. IRT assumes that individual test questions (referred to as "items" in IRT and "examples" here) have unique characteristics such as difficulty and discriminating power. These characteristics can be identified by fitting a joint model of human ability and examples characteristics to human response patterns to the test examples. Examples that do not fit the model can be removed and the remaining examples can be considered a scale to evaluate performance. IRT assumes that the probability of a correct answer is associated with both example characteristics and individual ability, and therefore a collection of examples of varying characteristics can determine an individual's ability overall.

IRT accounts for differences among examples when estimating a subject's ability. In addition, ability estimates from IRT are on the ability scale of the population used to estimate example parameters. For example, an estimated ability of 1.2 can be interpreted as 1.2 standard deviations above the average ability in this population. The traditional total number of correct responses generally does not have such quantitative meaning.

IRT has been widely used in educational testing. For example, it plays an instrumental role in the construction, evaluation or scoring of standardized tests such as Test of English as a Foreign Language (TOEFL), Graduate Record Examinations (GRE) and SAT.

1.3.1 IRT Models

The simplest IRT model assumes a single latent parameter for each example, b_i, corresponding to the example's difficulty, as well as a latent ability parameter for each model, θ_j. This is known as the one parameter logistic (1PL) model or the Rasch model.

The probability that model (or subject) j will answer example i correctly is:

$$p(y_{ij} = 1|\theta_j, b_i) = \frac{1}{1 + e^{-(\theta_j - b_i)}} \tag{1.5}$$

The probability that model j will answer example i incorrectly is:

$$p(y_{ij} = 0|\theta_j, b_i) = 1 - p(y_{ij} = 1|\theta_j, b_i) \tag{1.6}$$

With a 1PL model, there is an intuitive relationship between difficulty and ability. An example's difficulty value b can be thought of as the point on the ability scale where an individual (or model) has a 50% chance of answering correctly. Put another way, a model has a 50% chance of answering an example correctly when model ability is equal to example difficulty (if $\theta_j = b_i$ in Equation 1.5).

Another common model is the three parameter logistic model (3PL):

$$p_{ij}(\theta_j) = c_i + \frac{1 - c_i}{1 + e^{-a_i(\theta_j - b_i)}} \tag{1.7}$$

where a_i, b_i, and c_i are example parameters: the slope or discrimination parameter a_i is related to the steepness of the curve, the difficulty parameter b_i is the level of ability that produces a chance of correct response equal to the average of the upper and lower asymptotes, and the guessing parameter c_i is the lower asymptote of the ICC and the probability of guessing correctly. A two-parameter logistic (2PL) IRT model assumes that the guessing parameters are 0.

1.3.2 Parameter Estimation

The likelihood of a data set of response patterns Z from multiple subjects to a set of examples given the parameters Θ and B is:

$$p(Z|\Theta, B) = \prod_{j=1}^{J} \prod_{i=1}^{I} p(Z_{ij} = y_{ij}|\theta_j, b_i) \tag{1.8}$$

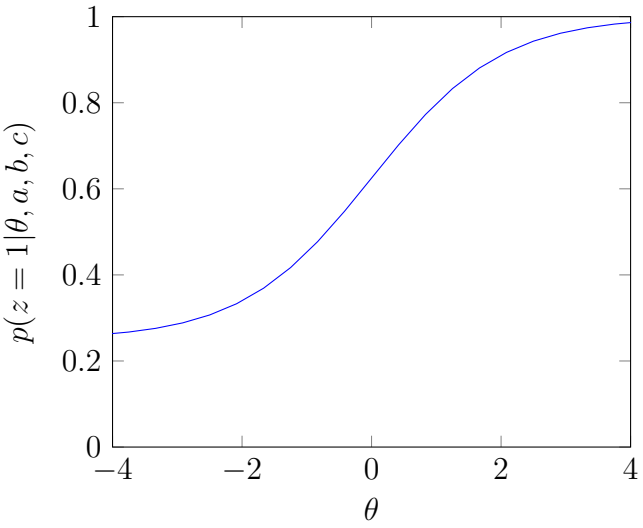


Figure 1.1: Example ICC for a "good" example, fit as part of a 3PL Model. For this example, $a_i = 1$, $b_i = 0$, and $c_i = 0.25$.

where $z_{ij} = 1$ if individual j answers example i correctly and $z_{ij} = 0$ if they do not.

The example parameters are typically estimated by marginal maximum likelihood (MML) via an Expectation-Maximization (EM) algorithm [Bock and Aitkin, 1981], in which subject parameters are considered random effects $\theta_i \sim N(0, \sigma_\theta^2)$ and marginalized out. Once example parameters are learned, subjects' θ parameters are scored typically with maximum a posteriori (MAP) estimation. IRT models are usually fitted to RPs of hundreds or thousands of human subjects, who usually answer at most 100 questions. Therefore the methods for fitting these models have not been scaled to huge data sets and large numbers of subjects (e.g. tens of thousands of machine learning models).

Figures 1.1 and 1.2 show examples of Item Characteristic Curves (ICCs) of two examples in a test set fit via a 3PL model. Figure 1.1 would be considered a "good" example, as there is a relatively steep slope distinguishing individuals that have a high probability of labeling the example correctly. Figure 1.2 would be considered a "bad" example. The slow increase in probability as ability increases indicates that this example is not useful for distinguishing between individuals. What's more, the very large guessing parameter indicates that even individuals with low latent ability have

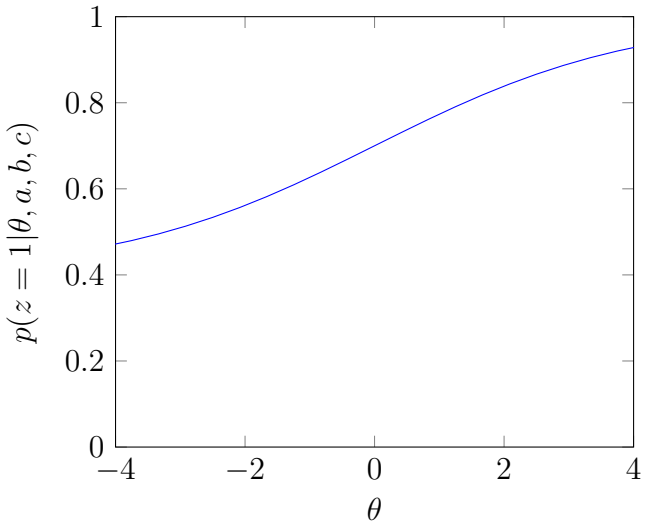


Figure 1.2: Example ICC for a "bad" example, fit as part of a 3PL Model. For this example, $a_i = 0.5$, $b_i = 0$, and $c_i = 0.4$.

a high probability of labeling the example correctly. The ICC plots the probability of a model labeling an example correctly as a function of latent ability. A good example should exhibit an ICC relatively steep slope increasing between ability levels -3 and 3, where most people are located, in order to have appropriate power to differentiate different levels of ability.

1.3.3 IRT with Variational Inference

Variation inference (VI) is a model fitting method that approximates an intractable posterior distribution in Bayesian inference by a simpler variational distribution. Prior work has compared VI methods with traditional IRT methods [Natesan et al., 2016] and found it effective, but was primarily concerned with fitting IRT models for human-scale data.

Bayesian methods in IRT assume that the individual θ and b parameters in Eq. (2) both follow Gaussian prior distributions and make inference through the resultant joint posterior distribution $\pi(\theta, b|Y)$. As this posterior is usually intractable, VI approximates it by the variational distribution:

$$q(\theta, b) = \prod_{j=1}^{J} \pi_j^{\theta}(\theta_j) \prod_{i=1}^{I} \pi_i^{b}(b_i) \tag{1.9}$$

Where $\pi_j^{\theta}()$ and $\pi_i^{b}()$ denotes different Gaussian densities for different parameters whose means and variances are determined by minimizing the KL-Divergence between $q(\theta, b)$ and $\pi(\theta, b|Y)$.

The choice of priors in Bayesian IRT can vary. Prior work has shown that vague and hierarchical priors are both effective [Natesan et al., 2016]. We experiment with both in this work. A vague prior assumes $\theta_j \sim N(0, 1)$ and $b_i \sim N(0, 10^3)$, where the large variance indicates a lack of information on the difficulty parameters. A hierarchical Bayesian model assumes

$$\theta_j \mid m_\theta, u_\theta \sim N(m_\theta, u_\theta^{-1})$$
$$b_i \mid m_b, u_b \sim N(m_b, u_b^{-1})$$
$$m_\theta, m_b \sim N(0, 10^6)$$
$$u_\theta, u_b \sim \Gamma(1, 1)$$

Our results for these two options were very similar, so we only report those for hierarchical priors.

1.3.4 Building IRT Test Sets

To identify the number of factors in an IRT model, the polychoric correlation matrix of the examples is calculated and its ordered eigenvalues are plotted. The number of factors is suggested by the number of large eigenvalues. It can be further established by fitting (see below) and comparing IRT models with different numbers of factors. Such comparison may use model selection indices such as AIC and CBIC and should also take into account the interpretablility of the loading pattern that links examples to factors.

An IRT model can be fit to data by marginal maximum likelihood method through an EM algorithm [Bock and Aitkin, 1981]. The marginal likelihood function is the probability to observe the current observed response patterns as a function of the example parameters with the persons' ability parameters integrated out as random effects. This function is maximized to produce estimates of the example parameters. For IRT models with more than one factor, the slope parameters (i.e. loadings) that relate examples and factors must be properly rotated [Browne, 2001] before they can be interpreted. Given the estimated example parameters, Bayesian estimates of the individual person's ability parameters are obtained with the standard normal prior distribution.

After determining the number of factors and fitting the model, the local independence assumption can be checked using the residuals of marginal responses of example pairs [Chen and Thissen, 1997] and the fit of the ICC for each example can be checked with item fit statistics [Orlando and Thissen, 2000]. If both tests are passed and all examples have proper discrimination power, then the set of examples is considered a calibrated measurement scale and the estimated example parameters can be further used to estimate an individual person's ability level.

1.3.5 Exploratory Model Fitting

Once a set of response patterns is gathered, it is not enough to simply fit an IRT model and use the result as your IRT test set. The first step is to identify a subset of examples that meet the underlying assumptions of IRT:

1. People differ from each other on an unobserved latent dimension of interest (usually called "ability")

2. The probability of correctly answering a particular example is a function of the latent ability dimension (the item characteristic curve, ICC)

3. Responses to individual examples are independent of each other for a given ability level of a person (the "local independence assumption")

4. Responses from different individuals are independent of each other.

In this section we describe the process of fitting an exploratory model. A number of software programs exist to automate portions of this process, in particular the mirt R package.

The first step is to confirm that there is a single underlying factor in the response pattern data set. If there are multiple latent factors, then a multi-factor model must be used, or the data must be split according to the latent factors to create multiple tests. To check the latent factors, you can plot the tetrachoric matrix to visualize the eigenvalues of the response pattern data. If there is a single large latent factor then you can proceed with a single factor model. This first step is crucial as it underlies the rest of the reasoning for building an IRT model. The goal is to develop a test that measures a latent ability parameter of some set of individuals for some task. If there are multiple latent factors in the data, then trying to learn a single latent θ will not accurately capture the data.

Once a single factor model has been confirmed as appropriate, the next step is to determine the most appropriate model given the characteristics of the examples. Is a 3 parameter logistic (3PL) model more appropriate than a 2 parameter logistic (2PL) model? That is, do we need to account for the guessing parameters for the examples in the data set? To do this one must first fit both 3PL and 2PL models (Chapter 2) and compare the model fits using traditional model fit statistics such as Akaike information criterion (AIC) [Akaike, 1974] or Bayesian Information Criterion (BIC) [Schwarz et al., 1978]. If a 2PL model is a better fit, than you can continue and not worry about the example guessing parameters. If the 3PL model is a better fit, the next step is to determine if, for each example in the response pattern set, the guessing parameter is significantly different from 0. For each example, if the guessing

parameter is not significantly different than 0 then a 2PL model is used. Therefore it is possible to construct a test model that is a combination of 2PL and 3PL models for each of the examples.

To identify the number of latent factors, a plot of eigenvalues of the tetrachoric correlation matrix can be inspected and a comparison between IRT models with different number of factors can be conducted. A target rotation [Browne, 2001] can be used to identify a meaningful loading pattern that associates factors and examples. If there are multiple latent factors present, the target rotation can be used to align the factors with specific sub-tasks. For example, in the case of NLI, if three latent factors are present, each factor can be interpreted as the ability of a user to recognize the correct relationship between the sentence pairs associated with that factor (e.g. contradiction).

1.3.6 Confirmatory Model Fitting

Once a model has been fit that best represents the response pattern data, it is important to confirm that the model did not overfit the data by conducting a confirmatory analysis. To do this, a new set of response patterns for the same set of examples are collected from a new population of test-takers. With the pre-fit example parameters, a new IRT model is fit to estimate θ and the model fit statistics are examined. If the fit statistics are reasonable, then the model is determined to be appropriate for the task. Otherwise, a new model must be fit.

1.3.7 Scoring

Estimating the ability of a model at a point in time is done with a "scoring" function. When example difficulties are known, model ability is estimated by maximizing the likelihood of the data given the response patterns and the example difficulties to obtain the ability estimate. All that is required is a single forward pass of the model on the data, as is typically done with a test or validation set.

$$Z_j = \forall_{y \in Y} \mathbf{I}[y_i = \hat{y}_i] \tag{1.10}$$

$$L(\theta_j | Z_j) = p(Z_j | \theta_j) \tag{1.11}$$

$$\hat{\theta}_j = \arg\max_{\theta_j} \prod_{i=1}^{I} p(z_{ij} = y_{ij} | \theta_j) \tag{1.12}$$

1.4 Related Work

1.4.1 Uncertainty in Machine Learning

There are several other areas of study regarding how best to use training data that are related to this work. Re-weighting or re-ordering training examples is a well-studied and related area of supervised learning. Often examples are re-weighted according to some notion of difficulty, or model uncertainty [Bengio et al., 2009, Chang et al., 2017]. In particular, the internal uncertainty of the model is used as the basis for selecting how training examples are weighted. For example, the history of model predictions for an example up to time $t-1$ can be used to estimate the model probability of labeling the example correctly [Chang et al., 2017]. However, model uncertainty is dependent upon the original data set the model was trained on, and is representative of uncertainty with respect to this particular model. This can be considered a local measure of uncertainty and may not be comparable across models.

This work is related to transfer learning and domain adaptation [Caruana, 1995, Bengio et al., 2011, Bengio, 2012], but with an important distinction. Transfer learning and domain adaptation repurpose representations learned for a source domain to facilitate learning in a target domain. We want to improve performance in the source domain by fine-tuning with data from the source domain with distributions over class labels. This work differs from domain adaptation and transfer learning in that we are not adding data from a different domain or applying a learned model to a new task. Instead, we are augmenting a single classification task by using a richer representation

of where the data lies within the class labels to inform training. The goal is that by fine tuning with a distribution over labels, a model will be less likely to overfit on a training set.

Prior work has considered IRT in the context of evaluating ML models using machine-generated [Martınez-Plumed et al., 2016] response patterns. In one study the authors attempted to fit IRT models using machine generated response patterns on small data sets (i.e. 200-300 examples), but obtained results that are difficult to interpret using the existing IRT assumptions [Martınez-Plumed et al., 2016]. To the best of our knowledge no one has attempted to fit IRT models using DNN-generated response patterns on large data sets.

1.4.2 Latent Modeling for Crowds

Prior work has considered modeling latent characteristics of examples and/or models. In particular, latent-variable models have been developed to identify low-quality annotators (*spammers*) [Hovy et al., 2013]. The proposed model assumes that an annotator either produces the correct label or guess randomly with a guessing parameter varying only across annotators. Other work used the Dawid & Skene model in which an annotator's response depends on both the true label and the annotator [Dawid and Skene, 1979, Passonneau and Carpenter, 2014]. In both models an annotator's response depends on an example only through its correct label. In contrast, IRT assumes a more sophisticated response mechanism involving both annotator qualities and example characteristics. To our knowledge we are the first to introduce IRT to NLP and to create a gold standard with the intention of comparing NLP applications to human intelligence.

The quality of crowdsourced data for linguistics research has been evaluated as well [Munro et al., 2010]. In that work the authors recreate classic linguistic studies and provide evaluation metrics for the obtained data. They compare crowd-generated

data with controlled experiments, whereas we use the crowd to identify data set examples for a discriminating test set for future evaluations. Identifying true labels via latent-trait models in the past has relied on a small number of annotators [Bruce and Wiebe, 1999]. That work uses 4 annotators at varying levels of expertise and does not consider the discriminating power of data set examples.

1.4.3 Soft Labels

Other work on modeling uncertainty in labels is Knowledge Distillation [Hinton et al., 2015]. In Knowledge Distillation, output probabilities of a complex expert model are used as input to a simpler model so the simpler model can learn to generalize based on the output weights of the expert model. The expectation is that how an expert model assigns output weights can be used to reduce overfitting in the simpler model. However with Knowledge Distillation, the expert model that is distilling its knowledge was still trained with a single class label as the gold standard, and the expert passes its uncertainty to the simpler model. In our work we capture uncertainty at the original training data, in order to induce generalization as part of the original training.

This work is also related to the idea of "crowd truth" and the CrowdTruth platform for collecting and using annotations from the crowd [Kajino et al., 2012, Inel et al., 2014]. The crowd truth assumption is that disagreement between annotators provides signal about data ambiguity and should be used in the learning process. CrowdTruth includes several metrics to calculate likelihoods of different events with regards to particular examples and particular annotators. In those cases, particularly with regards to annotators, the metrics are used to identify potential low-quality annotators for removal. We have a large number of annotations for each example (1000 annotations per example), and therefore we assume that any issues of annotator quality will be "drowned out" by the large number of annotations. Therefore we do not need to

identify and remove annotations, and instead can use raw annotation metrics instead of the CrowdTruth metrics. In addition this work is closely related to the idea of Label Distribution Learning (LDL) from Computer Vision (CV) [Geng, 2016]. For training and testing, LDL assumes that y is a probability distribution over labels. With LDL, the goal is to learn a distribution over labels. However in our case we would still like to learn a classifier that outputs a single class, while using the distribution over training labels as a measure of uncertainty in the data. We use the distribution over labels to represent the uncertainty associated with different examples in order to improve model training.

To the best of our knowledge this is the first work to use a subset of soft labeled data for fine-tuning, whereas previous work used an all-or-none approach (all hard or soft labels).

1.4.4 One-Shot Learning

One area of ML research in a similar category to the IRT work proposed here is one-shot learning. One-shot learning is an attempt to build ML models that can generalize after being trained on one or a few examples of a class as opposed to a large training set [Lake et al., 2013]. One-shot learning attempts to mimic human *learning behaviors* (i.e., generalization after being exposed to a small number of training examples) [Lake et al., 2013]. Our work instead looks at comparisons to human *performance*, where any learning (on the part of models) has been completed beforehand. Our goal is to analyze DNN models and training set variations as they affect ability in the context of IRT.

1.4.5 Curriculum Learning

Curriculum learning (CL) is a training procedure where models are trained to learn simple concepts before more complex concepts are introduced [Bengio et al., 2009]. CL training for neural networks can improve generalization and speed up convergence. In

curriculum learning the difficulty of examples is typically assigned based on heuristics of the data (e.g. the number of sides of a shape). IRT models directly estimate difficulty from the responses of human or machine test-takers themselves instead of relying on heuristics. Self-paced learning and the Leitner method use model performance to estimate difficulties, but are restricted to a single model's performance, not a more global notion of difficulty [Kumar et al., 2010, Amiri et al., 2018].

Since its original proposal, curriculum learning has become a well-studied area of machine learning [Bengio et al., 2009]. The primary focus has been on developing new heuristics to identify easy and difficult examples in order to build a curriculum. Originally, curriculum learning methods were evaluated on toy data sets with heuristic measures of difficulty [Bengio et al., 2009]. For example, on a shapes data set, shapes with more sides were considered more difficult than shapes with fewer sides. Similarly, sentences with more words were considered more difficult than sentences with fewer words.

Recent work has shown that spaced repetition strategies (SR) can be effective for improving model performance [Amiri et al., 2017, Amiri, 2019]. Instead of using a traditional curriculum learning setup, spaced repetition bins examples based on estimated difficulty. The bins are shown to the model at differing intervals so that more difficult examples are seen more frequently than easier examples. This method has been shown to be effective for human learning, and results demonstrate effectiveness on NLP tasks as well. Similarly to traditional curriculum learning frameworks, SR uses model-dependent heuristics for difficulty and rigid schedulers to determine when training examples should be re-introduced to the learner.

Recent work has shown that measuring model competency during training to determine which examples to include at a training epoch further improves performance by matching data to model competency [Platanios et al., 2019]. However, in that work the model of competency is based on a heuristic rate of knowledge acquisition, and

does not actually measure model competency. To the best of our knowledge this is the first work to match model ability at a point in training with appropriate training data in a curriculum learning framework.

There has been recent work investigating the theory behind curriculum learning [Weinshall et al., 2018, Hacohen and Weinshall, 2019], particularly around trying to define an ideal curriculum. The authors explicitly identify the two key aspects of curriculum learning, namely "sorting by difficulty" and "pacing." curriculum learning theoretically leads to a steeper optimization landscape (i.e. faster learning) while keeping the same global minimum of the task without curriculum learning. In that work there is still a reliance on "pacing functions" as opposed to an actual assessment of model ability at a point in time.

Hacohen and Weinshall also demonstrated a key distinction between curriculum learning and similar methods such as self-paced learning [Kumar et al., 2010], hard example mining [Shrivastava et al., 2016], and boosting [Freund and Schapire, 1997]: namely that the former considers difficulty with respect to the final hypothesis space (i.e. a model trained on the full data set) while the later methods consider ranking examples according to how difficult the current model determines them to be [Hacohen and Weinshall, 2019]. In this work we bridge the gap between these methods by probing model ability at the current point in training and using this estimated ability to identify appropriate training examples in terms of global difficulty.

CHAPTER 2

BUILDING NLP TEST SETS WITH ITEM RESPONSE THEORY

In this chapter we will demonstrate the usefulness of using Item Response Theory (IRT) for building test sets. IRT has been used to build test sets for many years and in many contexts, and the methodology is well-established. Here, we apply these methods to natural language processing for the first time with two representative tasks: natural language inference (NLI) and sentiment analysis (SA).

The rest of this chapter is structured as follows: we first describe IRT and the process of building a test set with IRT in detail. Then, we describes the data collection, model fitting, and evaluation of the IRT NLP test sets. We then demonstrate the use of the test sets on deep learning models for each NLP task.

2.1 Item Response Theory for Test Set Generation

The process of building an IRT test set can be broken down into three parts: response pattern collection, exploratory model fitting, and confirmatory model fitting. We will describe each of these steps generally here, with specifics in the followings sections for the NLP and EHR comprehension tests, respectively. To begin one must first have a pool of examples from which the test set will be obtained. This could be a large pool of previously written questions, or a data set for a specific task in the context of NLP. For this example pool, a large IRT model is fit and examples are removed that do not fit, until you are left with a subset of examples that can estimate the latent dimension well.

Throughout this chapter the IRT model under consideration is the three parameter logistic (3PL) model, which was introduced in Section 1.3. Recall that the 3PL model estimates the probability that model j will answer example i correctly, given model j's latent ability θ_j and example i's discriminatory parameter a_i, difficulty b_i, and guessing parameter c_i:

$$p_{ij}(\theta_j) = c_i + \frac{1 - c_i}{1 + e^{-a_i(\theta_j - b_i)}} \tag{2.1}$$

2.1.1 Gathering Response Patterns

Before building an IRT test set, there must first be some example pool from which a subset can be extracted as an IRT test. This pool of examples typically consists of questions that seem appropriate for measuring the desired trait, but have not yet been validated. For example, for the SAT there is a pool of examples that have been written as candidates for inclusion for the test. These examples are included in the test periodically and their latent characteristics are evaluated to determine if they should be included in the test [Carlson and von Davier, 2013].

To learn latent example parameters for a test set, one requires data. Specifically, it is necessary to first gather a large number of graded responses to the examples in the example pool in order to fit the IRT model. Following §1.1, let D_{pool} be the set of examples in the example pool under consideration for inclusion, where $X^{D_{pool}}$ and $Y^{D_{pool}}$ are the features and gold-standard labels associated with the examples in the pool, respectively. For some set of models J, let $\hat{y}_{ij}$ be model j's labeling of example i. Model j's response pattern Z_j is defined as the sequence of model j's provided labels, graded correct or incorrect against the gold standard label:

$$Z^j = \{\forall y_i \in Y, \mathbb{I}[y_{ij} = y_i^*]\} \tag{2.2}$$

where $\mathbb{I}[]$ is the indicator function, which evaluates to 1 when the expression is true and evaluates to 0 when the expression is false.

In a typical IRT testing scenario, response patterns are gathered from human subjects for a specific task. For example, new questions on the SAT are added to the test on a trial basis, and responses from students are gathered as they take the full test, and new questions are evaluated with respect to the existing test [Carlson and von Davier, 2013]. In other cases, a target population is identified and given the preliminary test questions, from which the IRT test set is identified. For example, a test of cancer patients was developed from response patterns taken from cancer patients [Mazor et al., 2012b, Mazor et al., 2012a]. However in our work, response patterns are gathered using crowdsourcing workers, specifically those on the Amazon Mechanical Turk (AMT) crowdsourcing platform.

AMT is an online microtask crowdsourcing platform where individuals (called Turkers) perform Human Intelligence Tasks (HITs) in exchange for payment. HITs are usually pieces of larger, more complex tasks that are have been broken up into multiple, smaller subtasks. AMT and other crowdsourcing platforms are used to build large corpora of human-labeled data at low cost compared to using expert annotators [Snow et al., 2008, Sabou et al., 2012]. Researchers' projects have used AMT to complete a variety of tasks [Demartini et al., 2012, Zhai et al., 2013]. Recent research has shown that AMT and other crowdsourcing platforms can be used to generate corpora for clinical natural language processing and disease mention annotation [Zhai et al., 2013,Good et al., 2015]. AMT was used to detect errors in a medical ontology and found that the crowd was as effective as domain experts [Mortensen et al., 2015]. In addition, AMT workers have been used to identify disease mentions in PubMed abstracts [Good et al., 2015] and rank Adverse Drug Reactions in order of severity [Gottlieb et al., 2015] with good results.

In order to ensure that the data gathered from the AMT Turkers was reliable, we included a number of quality control mechanisms in each of our tests:

1. AMT task access was restricted to individuals located in the United States, as a proxy for requiring English speakers

2. Tasks were only available to Turkers who have a prior task approval rate of 97% or higher

3. Within each task periodic attention-check questions were included, designed to ensure that the Turkers were paying attention and answering the questions to the best of their ability. Responses where the attention-check questions were answered incorrectly were removed.

For each of our IRT tests, we gathered enough response patterns based on the size of our example banks to ensure that the fit IRT models were reliable. While there is no set standard for sample sizes in IRT models, this sample size satisfies the standards based on the non-central χ^2 distribution [MacCallum et al., 1996] used when comparing two multidimensional IRT models. This sample size is also appropriate for tests of example fit and local dependence that are based on small contingency tables. To identify appropriate examples for the test sets we conducted both exploratory (§1.3.5) and confirmatory (§1.3.6) analysis of the response pattern data.

We built a unidimensional IRT model for each set of examples associated with a single factor. We fit and compared one- and two-factor 3PL models to confirm the unidimensional structure underlying these examples, assuming the possible presence of guessing in people's responses. We further tested the guessing parameter of each example in the one factor 3PL model. If it was not significantly different from 0, a 2PL ICC was used for that particular example.

Once an appropriate model structure was determined, individual examples were evaluated for goodness of fit within the model. If an example was deemed to fit the

ICC poorly or to give rise to local dependence, it was removed for violating model assumptions. Furthermore, if the ICC of an example was too flat, it was removed for low discriminating power between ability levels. The model was then refit with the remaining examples. This iterative process continued until no example could be removed (2 to 6 iterations depending on how many examples were removed from each set).

2.2 Evaluating Natural Language Processing Models

Evaluation of NLP methods requires testing against a previously vetted gold-standard test set and reporting standard metrics (accuracy/precision/recall/F1). The current assumption is that all examples in the test set are equal with regards to difficulty and discriminating power. However IRT can be used as an alternative means for gold-standard test-set generation and NLP method evaluation. IRT is able to describe characteristics of individual examples - their difficulty and discriminating power - and is able to account for these characteristics in estimating latent ability for an NLP task. We demonstrate IRT by generating a gold-standard test set for natural language inference (NLI) and sentiment analysis (SA). By collecting a large number of human responses and fitting our IRT model, we show that our IRT model compares NLP systems with the performance in a population and is able to score differently from the standard evaluation metrics. We show that a high accuracy score does not always imply a high IRT score, which depends on the example characteristics and the response pattern.

Our aim is to build an intelligent evaluation metric to measure performance for NLP tasks. With IRT one can identify an appropriate set of examples to measure ability in relation to the overall human population as scored by an IRT model. This process serves two purposes: (i) to identify individual examples appropriate for a test set that measures ability on a particular task, and (ii) to use the resulting set of

examples as an evaluation set in its own right, to measure the ability of future subjects (or NLP models) for the same task. These evaluation sets can measure the ability of an NLP system with a small number of examples, leaving a larger percentage of a data set for training.

2.2.1 Tasks under Consideration

2.2.1.1 Natural Language Inference

NLI was introduced to standardize the challenge of accounting for semantic variation when building models for a number of NLP applications [Dagan et al., 2006]. NLI defines a directional relationship between a pair of sentences, the text (T) and the hypothesis (H). T entails H if a human that has read T would infer that H is true. If a human would infer that H is false, then H contradicts T. If the two sentences are unrelated, then the pair are said to be neutral. Table 2.3 shows examples of T-H pairs and their respective classifications. Recent state-of-the-art systems for NLI require a large amount of feature engineering and specialization to achieve high performance [Beltagy et al., 2016, Lai and Hockenmaier, 2014, Jimenez et al., 2014].

A number of gold-standard data sets are available for NLI [Marelli et al., 2014, Young et al., 2014, Levy et al., 2014]. We consider the Stanford Natural Language Inference (SNLI) data set [Bowman et al., 2015]. SNLI is an English-language natural language inference (NLI) data set that consists of human-generated sentence pairs and NLI labels (entailment, contradiction, or neutral). SNLI examples were generated using only human-generated sentences to mitigate the problem of poor data that was being used to build models for NLI. In addition, SNLI included a quality control assessment of a sampled portion of the data set (about 10%, 56,951 sentence pairs). This data was provided to 4 additional AMT users to provide labels (entailment, contradiction, neutral) for the sentence pairs. If at least 3 of the 5 annotators (the original annotator

and 4 additional annotators) agreed on a label the example was retained. Most of the examples (98%) received a gold-standard label.

SNLI is an order of magnitude larger than previously available NLI data sets (550k train/10k dev/10k test), and consists entirely of human-generated P-H pairs. SNLI is evenly split across three labels: entailment, contradiction, and neutral.

Amazon Mechanical Turk (AMT) users were shown a caption that was taken from the Flickr30k corpus [Young et al., 2014] and told that the caption was associated with a photo. The users were not shown the corresponding photo. They were then asked to write three alternate captions that could describe the photo: (i) one that is definitely true, (ii) one that might be true, and (iii) one that is definitely false. These newly generated sentences were then combined with the original caption to create entailment, neutral, and contradiction sentence pairs, respectively.

2.2.1.2 Sentiment Analysis

The Stanford Sentiment Treebank (SSTB) [Socher et al., 2013] is a collection of English text snippets extracted from movie reviews with fine-grained sentiment annotations (very negative, negative, neutral, positive, very positive). SSTB includes sentence- and phrase-level sentiment labels for 11,000 sentences (215,000 phrases). SNLI is large, well-studied, and often used as a benchmark for new NLP models for NLI. The data set consists of 67k/873/1.8k training/validation/testing examples.

2.2.2 Example Selection

We collected and evaluated a random selection from the SNLI NLI data set (GS_{NLI}) to build our IRT models. We first randomly selected a subset of GS_{NLI}, and then used the sample in an AMT Human Intelligence Task (HIT) to collect more labels for each text-hypothesis pair. We then applied IRT to evaluate the quality of the examples and used the final IRT models to create evaluation sets (GS_{IRT}) to measure ability for NLI.

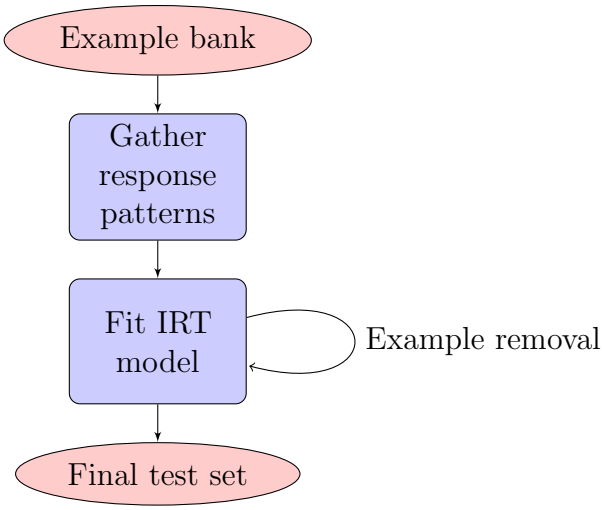


Figure 2.1: High level overview of the test set construction process with IRT.

We selected a subset of GS_{NLI} to use as an examination set according to the following steps: (1) Identify all "quality-control" examples from GS_{NLI} as described in 2.2.1.1, (2) Split this section of the data according to the number of users that agreed on the eventual gold standard label, (3) Randomly select 30 entailment sentence pairs, 30 neutral pairs, and 30 contradiction pairs from each of the 4-annotator gold standard (4GS) and 5-annotator gold standard (5GS) sets to obtain two sets of 90 sentence pairs.

90 sentence pairs for 4GS and 5GS were sampled so that the annotation (supplying 90 labels) could be completed in a reasonably short amount of time during which users remained engaged. We selected examples from 4GS and 5GS because both groups are considered high quality for NLI. We evaluated the selected 180 sentence pairs using the model provided with the original data set [Bowman et al., 2015] and found that accuracy scores were similar compared to performance on the SNLI test set.

2.2.3 AMT Annotation

For consistency we designed our AMT HIT to match the process used to validate the SNLI quality control examples [Bowman et al., 2015] and to generate labels

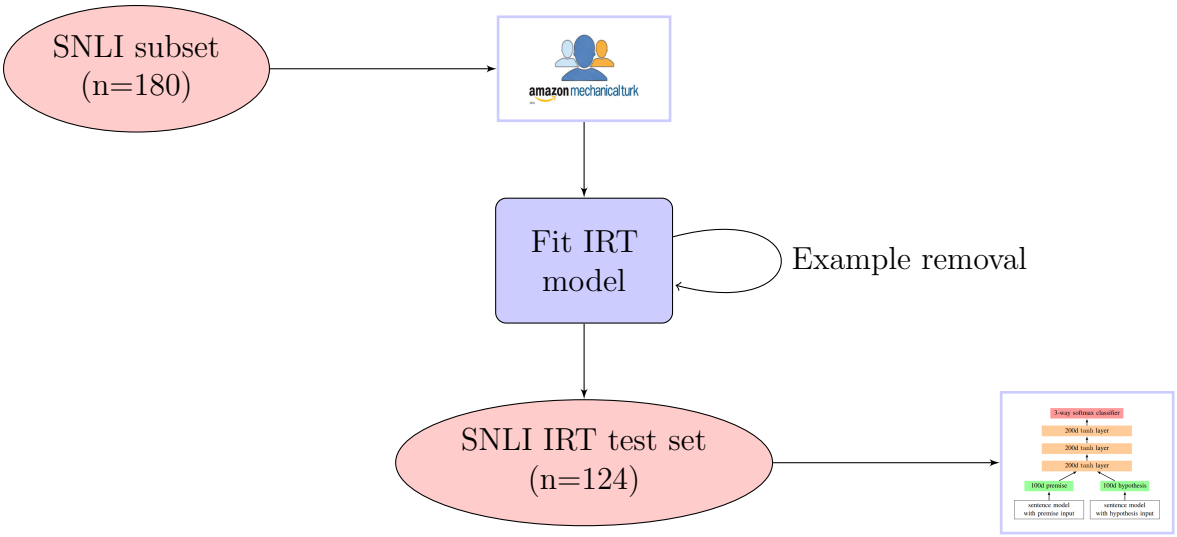


Figure 2.2: Building an IRT test set for the SNLI data set. Response patterns were obtained from Amazon Mechanical Turk workers (Turkers) and processed using IRT. A subset of examples were retained following analysis as the final test set. The test set can then be administered to a trained DNN model.

for the SICK NLI data set [Marelli et al., 2014]. Each AMT user was shown 90 premise-hypothesis pairs (either the full 5GS or 4GS set) one pair at a time, and was asked to choose the appropriate label for each. Each user was presented with the full set, as opposed to one-label subsets (e.g. just the entailment pairs) in order to avoid a user simply answering with the same label for each example.

For each 90 sentence-pair set (5GS and 4GS), we collected annotations from 1000 AMT users, resulting in 1000 label annotations for each of the 180 sentence pairs according to the standards based on the non-central χ^2 distribution [MacCallum et al., 1996] used when comparing two multidimensional IRT models (Section 2.1.1).

We applied a set of quality control checks (Section 2.1.1) to ensure that the labels gathered were of a high quality. After removing individuals that failed the attention-check, we retained 976 labels for each example in the 4GS set and 983 labels for each example in the 5GS set. Average time spent for each task was roughly 30 minutes, a reasonable amount for AMT users.

2.2.4 Statistical Analysis

We performed the exploratory model analysis on the gathered data (Section 1.3.5). Data collected for 4GS and 5GS were analyzed separately. For both sets of examples, the number of factors was identified by a plot of eigenvalues of the 90 x 90 tetrachoric correlation matrix and by a further comparison between IRT models with different number of factors. A target rotation [Browne, 2001] was used to identify a meaningful loading pattern that associates factors and examples. Each factor could then be interpreted as the ability of a user to recognize the correct relationship between the sentence pairs associated with that factor (e.g. contradiction).

We built a unidimensional IRT model for each set of examples associated with a single factor. We fit and compared one- and two-factor 3PL models to confirm the unidimensional structure underlying these examples, assuming the possible presence of guessing in people's responses. We further tested the guessing parameter of each example in the one factor 3PL model. If it was not significantly different from 0, a 2PL ICC was used for that particular example.

Once an appropriate model structure was determined, individual examples were evaluated for goodness of fit within the model. If an example was deemed to fit the ICC poorly or to give rise to local dependence, it was removed for violating model assumptions. Furthermore, if the ICC of an example was too flat, it was removed for low discriminating power between ability levels. The model was then refit with the remaining examples. This iterative process continued until no example could be removed (2 to 6 iterations depending on how many examples were removed from each set).

The remaining examples make up our final test set (GS_{IRT}), which is a calibrated scale of ability to correctly identify the relationship between the two sentence pairs. Parameters of these examples were estimated as part of the IRT model and the set of

	4GS	5GS	Overall
Pairs with majority agreement	95.6%	96.7%	96.1%
Pairs with supermajority agreement	61.1%	82.2%	71.7%
Individual Label = gold label	73.2%	82.3%	77.7%
New gold label = original gold label	81.1%	93.3%	87.2%

Table 2.1: Summary statistics from the AMT HITs.

examples can be used as an evaluation scale to estimate ability of test-takers or NLI systems. We used the *mirt* R package [Chalmers et al., 2012] for our analyses.

2.2.5 Response Statistics

Table 2.1 lists key statistics from the AMT HITs. Most of the sampled sentence pairs resulted in a gold standard label being identified via a majority vote. Due to the large number of individuals providing labels during the HIT, we also wanted to see if a gold standard label could be determined via a two-thirds supermajority vote. We found that 28.3% of the sentence pairs did not have a supermajority gold label. This highlights the ambiguity associated with identifying entailment.

We believe that the examples selected for analysis are appropriate for our task in that we chose high-quality examples, where at least 4 annotators selected the same label, indicating a strong level of agreement (Section 2.2.2). We argue that our sample is a high-quality portion of the data set, and further analysis of examples where the gold-standard label was only selected by 3 annotators originally would result in lower levels of agreement.

Table 2.2 shows that the level of agreement as measured by the Fleiss' κ score is much lower when the number of annotators is increased, particularly for the 4GS set of sentence pairs, as compared to scores noted in [Bowman et al., 2015]. The decrease in agreement is particularly large with regard to contradiction. This could occur for a number of reasons. Recognizing entailment is an inherently difficult task, and classifying a correct label, particularly for contradiction and neutral, can be

Data Set	Fleiss' κ	Originally Reported Agreement
SNLI 4GS Contradiction	0.37	0.77 [Bowman et al., 2015]
SNLI 5GS Contradiction	0.59	
SNLI 4GS Entailment	0.48	0.72 [Bowman et al., 2015]
SNLI 5GS Entailment	0.63	
SNLI 4GS Neutral	0.41	0.6 [Bowman et al., 2015]
SNLI 5GS Neutral	0.54	
SSTB	0.52	n/a

Table 2.2: Fleiss' κ scores for the NLI and SA annotations collected from AMT. Original label-level agreement scores for SNLI are also reported. Inter-annotator agreement was not reported during SSTB collection.

difficult due to an individual's interpretation of the sentences and assumptions that an individual makes about the key facts of each sentence (e.g. coreference). It may also be the case that the individuals tasked with creating the sentence pairs on AMT created sentences that appeared to contradict a premise text, but can be interpreted differently given a different context.

Inter-rater reliability scores for the collected annotations are shown in Table 2.2. Human annotations for the SA annotations were converted to binary before calculating the agreement. The agreement scores are in the range of 0.4 to 0.6 which is considered moderate agreement [Landis and Koch, 1977]. With the large number of annotators it is to be expected that there is some disagreement in the labels. However this disagreement can be interpreted as varying difficulty of the examples, which is what we expect when we fit the IRT models. In addition, when the SNLI data set was originally collected, Turkers were instructed to label the premise-hypothesis relationship with the understanding that the pair related to an (unseen) photo. The instructions in our task were more general, and referred to the relationship between the two sentences generally. Therefore there is more room for disagreement between Turkers, but since the release of SNLI the concept of the sentence pairs referring to a photo has not been applied to downstream learning tasks, so we feel that the general labeling task is more representative of the typical use case for the data set.

Before fitting the IRT models we performed a visual inspection of the 180 sentence pairs and removed examples clearly not suitable for an evaluation scale due to syntactic or semantic discrepancies. For example, example 10 in Table 2.3 was removed from the 5GS contradiction set for semantic reasons. While many people would agree that the statement is a contradiction due to the difference between football and soccer, individuals from outside the U.S. would possible consider the two to be synonyms and classify this as entailment. Six such pairs were identified and removed from the set of 180 examples, leaving 174 examples for IRT model-fitting.

2.2.6 IRT Evaluation

We used the methods described in Section 2.2.4 to build IRT models to evaluate performance according to the NLI task. For both 4GS and 5GS examples three factors were identified, each related to examples for the three GS_{RTE} labels (entailment, contradiction, neutral). This suggests that examples with the same GS_{RTE} label within each set defines a separate ability. In the subsequent steps, examples with different labels were analyzed separately. After analysis, we were left with a subset of the 180 originally selected examples. Refer to Table 2.3 for examples of the retained and removed examples based on the IRT analysis. We retained 124 of the 180 examples (68.9%). We were able to retain more examples from the 5GS data sets (76 out of 90 - 84%) than from the 4GS data sets (48 out of 90 - 53.5%). Examples that measure contradiction were retained at the lowest rate for both 4GS and 5GS data sets (66% in both cases). For the 4GS entailment examples, our analysis found that a one-factor model did not fit the data, and a two-factor model failed to yield an interpretable loading pattern after rotation. We were unable to build an IRT model that accurately modeled ability to recognize entailment with the obtained response patterns. As a result, no examples from the 4GS entailment set were retained.

Group	Label	Text
Retained		
4GS	Neutral	1. Premise: A toddler playing with a toy car next to a dog Hypothesis: A toddler plays with toy cars while his dog sleeps
	Contradiction	2. Premise: People were watching the tournament in the stadium Hypothesis: The people are sitting outside on the grass
5GS	Contradiction	3. Premise: A person is shoveling snow Hypothesis: It rained today
	Neutral	4 Premise: Two girls on a bridge dancing with the city skyline in the background Hypothesis: The girls are sisters.
	Entailment	5. Premise: A woman is kneeling on the ground taking a photograph Hypothesis: A picture is being snapped
Removed		
4GS	Neutral	6. Premise: Two men and one woman are dressed in costume hats Hypothesis: The people are swingers
	Contradiction	7. Premise: Man sweeping trash outside a large statue Hypothesis: A man is on vacation
	Entailment	8. Premise: A couple is back to back in formal attire Hypothesis: Two people are facing away from each other
	Entailment	9. Premise: A man on stilts in a purple, yellow and white costume Hypothesis: A man is performing on stilts
5GS	Contradiction	10. Premise: A group of soccer players are grabbing onto each other as they go for the ball Hypothesis: A group of football players are playing a game
	Neutral	11. Premise: Football players stand at the line of scrimmage Hypothesis: The players are in uniform
	Entailment	12. Premise: Man in uniform waiting on a wall Hypothesis: Near a wall, a man in uniform is waiting

Table 2.3: Examples of retained & removed sentence pairs. The selection is not based on right/wrong labels but based on IRT model fitting and example elimination process. Note that no 4GS entailment examples were retained (Section 2.2.6)

Figure 2.3 plots the empirical spline-smoothed ICC of one example (Table 2.3, example 9) with its estimated response curve. The empirical ICC plots the probability values observed in the data for the estimated latent θ values, which is then smoothed to approximate a continuous function. The ICC is not continuously increasing, and thus a logistic function is not appropriate. This example was spotted for poor fit and removed. Figure 2.4 shows a comparison between the ICC plot of a retained example (Table 2.3, example 4) and the ICC of a removed example (Table 2.3, example 8). Note that the

removed example has an ICC that is very flat between -3 and 3. This example cannot discriminate individuals at any common level of ability and thus is not useful.

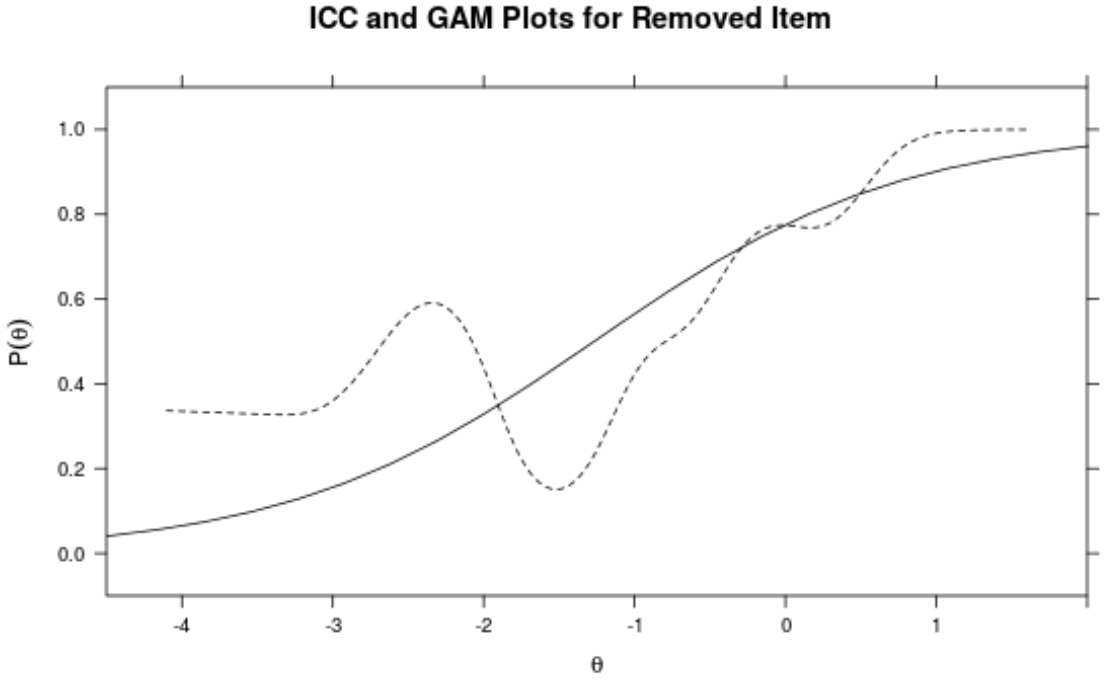


Figure 2.3: Estimated (solid) and actual (dotted) response curves for a removed example.

The examples retained for each factor can be considered as an evaluation scale that measures a single ability of an individual test-taker. As each factor is associated with a separate gold-standard label, each factor (θ) is a person's ability to correctly classify the relationship between the text and hypothesis for one such label (e.g. entailment).

2.2.7 Example Parameter Estimation

Parameter estimates of retained examples for each label are summarized in Table 2.4, and show that all parameters fall within reasonable ranges. All retained examples have 2PL ICCs, suggesting no significant guessing. Difficulty parameters of most examples are negative, suggesting that an average AMT user has at least 50% chance to answer these examples correctly. This low range of example difficulty (relative to a

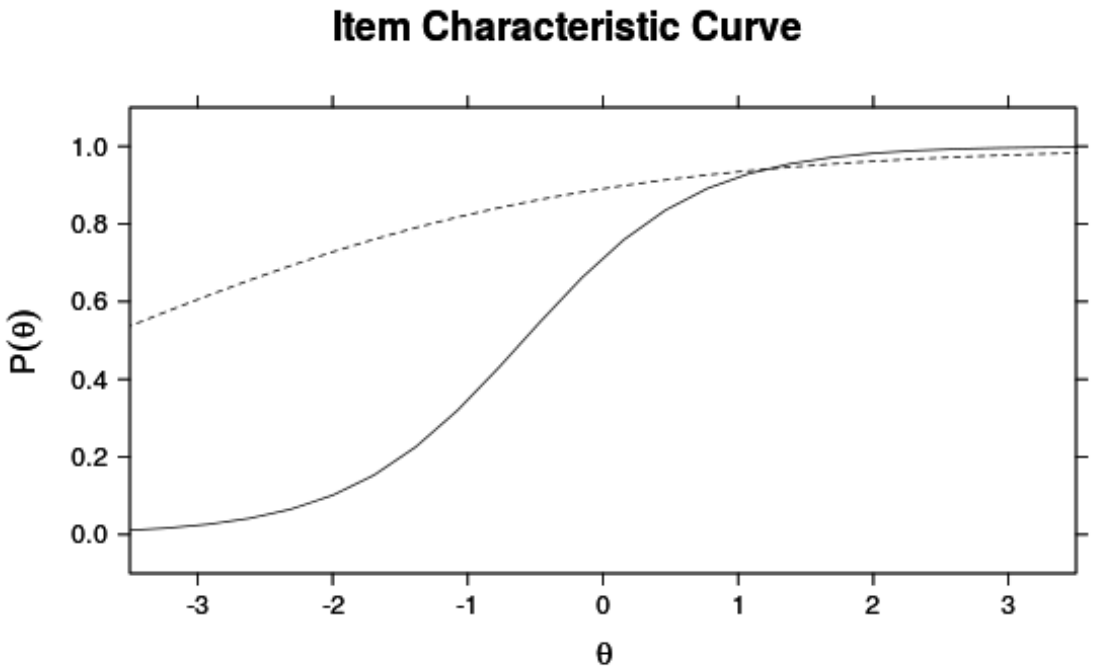


Figure 2.4: ICCs for retained (solid) and removed (dotted) examples.

human population) is appropriate for the evaluation of NLP systems. Examples in each scale have a wide range of difficulty and discrimination power.

		Difficulty		Slope	
	Example Set	**Min.**	**Max.**	**Min.**	**Max.**
5GS	Contradiction	-2.765	0.704	0.846	2.731
	Entailment	-3.253	-1.898	0.78	2.61
	Neutral	-2.082	-0.555	1.271	3.598
4GS	Contradiction	-1.829	1.283	0.888	2.753
	Neutral	-2.148	0.386	1.133	3.313

Table 2.4: Parameter estimates of the retained examples

With IRT one can use the heterogeneity of examples to properly account for such differences in the estimation of a test-taker's ability. Figure 2.5 plots the estimated ability of each AMT user from IRT against their total number of correct responses to the retained examples in the 4GS contradiction example set. The two estimates

of ability differ in many aspects. First, test-takers with the same total score may differ in their IRT score because they have different response patterns (i.e. they made mistakes on different examples), showing that IRT is able to account for differences among examples. Second, despite a rough monotonic trend between the two scores, people with a higher number of correct responses may have a lower ability estimate from IRT.

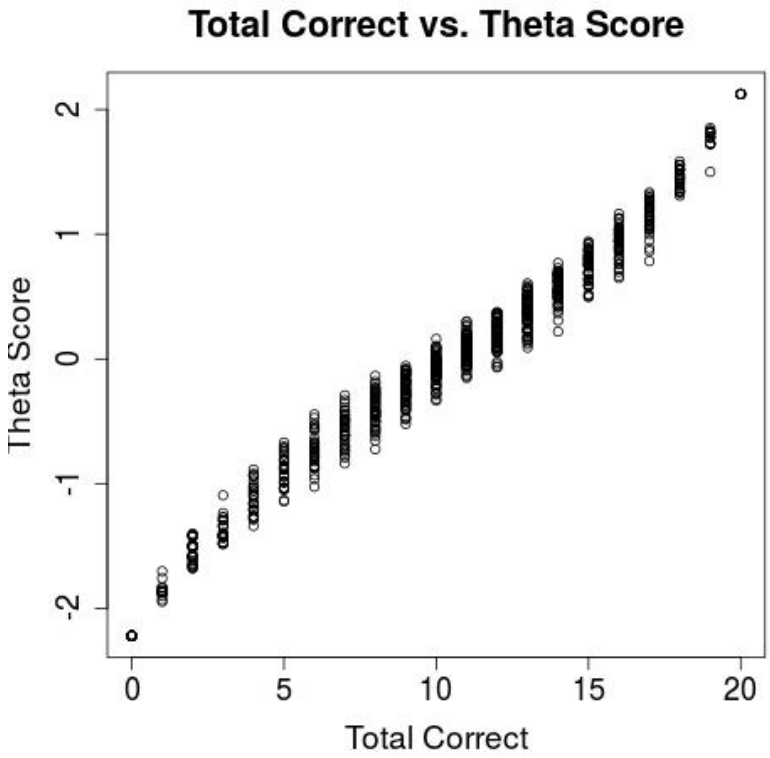


Figure 2.5: Plot of total correct answers vs. IRT scores.

We can extend this analysis to the case of NLI systems, and use the newly constructed scales to evaluate NLI systems. A system could be trained on an existing data set and then evaluated using the retained examples from the IRT models to estimate a new ability score. This score would be a measurement of how well the system performed with respect to the human population used to fit the model. With

this approach, larger sections of data sets can be devoted to training, with a small portion held out to build an IRT model that can be used for evaluation.

2.2.8 Application to an NLI System

As a demonstration, we evaluate the LSTM model presented in [Bowman et al., 2015] with the examples in our IRT evaluation scales. The goal here is not to achieve state of the art performance for the NLI task, but rather to demonstrate a use case of the IRT test sets on an existing model. In addition to the theta scores, we calculate accuracy for the binary classification task of identifying the correct label for all examples eligible for each subset in Table 5 (e.g. all test examples where 5 of 5 annotators labeled the example as *entailment* for 5GS). Note that these accuracy metrics are for subsets of the SNLI test set used for binary classifications and therefore do not compare with the standard SNLI test set accuracy measures. The theta scores from IRT in Table 2.5 show that, compared to AMT users, the system performed well above average for contradiction examples compared to human performance, and performed around the average for entailment and neutral examples. For both the neutral and contradiction examples, the theta scores are similar across the 4GS and 5GS sets, whereas the accuracy of the more difficult 4GS examples is consistently lower. This clearly demonstrates the advantage of IRT to account for example characteristics in its ability estimates. For 5GS the theta score and accuracy for 5GS entailment show that high accuracy does not necessarily mean that performance is above average when compared to human performance.

Majority vote validation of a gold standard has been in common use since the inception of NLP. It is easy to implement and evaluate, and allows for disagreements between annotators as long as one choice hits a certain threshold, usually 50% agreement. However, many factors may contribute to a majority vote. For example, an "easy" example with a majority vote may not be useful for separating the performance

	Example Set	Theta Score (Percentile)	Test Acc.
5GS	Entailment	-0.133 (44.83%)	96.5%
	Contradiction	1.539 (93.82%)	87.9%
	Neutral	0.423 (66.28%)	88%
4GS	Contradiction	1.777 (96.25%)	78.9%
	Neutral	0.441 (67%)	83%

Table 2.5: Theta scores and area under curve percentiles for LSTM trained on SNLI and tested on GS_{IRT}. We also report the accuracy for the same LSTM tested on all SNLI quality control examples (see Section 2.2.2). All performance is based on binary classification for each label.

of NLP systems. By using a limited number of annotators there is a risk of bias or uncertainty influencing the evaluation.

As NLP systems have become more sophisticated, sophisticated methodologies are required to compare their performance. One approach to create an intelligent gold standard is to use IRT to build models to scale performance on a small section of examples with respect to the tested population. IRT models can identify data set examples with different difficulty levels and discrimination powers based on human responses, and identify examples that are not appropriate as scale examples for evaluation. The resulting small set of examples can be used as a scale to score an individual or NLP system. This leaves a higher percentage of a data set to be used in the training of the system, while still having a valuable metric for testing.

Our current study uses the original GS_{NLI} labels as answer keys to define response patterns. A drawback of this is that our analysis depends on the validity of the original GS_{NLI} labels. However, IRT was still able to identify a final set of examples and provide their meaningful characteristics, showing the robustness of this approach.

2.3 Sentiment Analysis

For SA, we collected a new data set of labels for 134 examples randomly selected from the Stanford Sentiment Treebank (SSTB) [Socher et al., 2013], using a similar

AMT setup as for NLI. For each randomly selected example, we had 1000 Turkers label the sentence as very negative, negative, neutral, positive, or very positive. We converted these responses to binary positive/negative labels and fit a new IRT 3PL model using the *mirt* R package [Chalmers et al., 2012]. Very negative and negative labels were binned together, and neutral, positive, and very positive were binned together. To build the SA test set, the same procedure was followed as for the NLI test sets. First, the tetrachoric matrix was inspected to identify the number of latent factors. Single-factor and two-factor models were fit and compared to determine the latent structure. After confirming that a single-factor model was appropriate, each example was tested for goodness-of-fit, and removed if the example was a poor fit for the model. At the end of this process, 77 examples were retained and 54 were removed. Once the test was built, we used it to evaluate the same LSTM architecture as in the NLI task. We trained the LSTM model on the full SSTB training set (M1) and also on a training set where we randomly sampled two-thirds of the training data (M2). Test set output showed that more training data leads to higher performance (Table 2.6), as is expected. However we again see that by using the IRT test set model performance is more clearly delineated than with raw accuracy. The model trained with the full training set (M1) is significantly better than M2 with respect to the human population of Turkers. This result echoes that of the NLI test sets. It indicates that raw accuracy is not enough to accurately measure the performance of these models and progress for the task. Accuracy scores for SA have been approaching 99% with the most recent DNN architectures, however that does not mean that the task is solved. Instead, new data sets are required that measure difficult cases of SA.

2.4 Conclusion

We have introduced Item Response Theory from psychometrics as an alternative method for generating gold-standard evaluation data sets. Fitting IRT models allows

Model	θ (Percentile)	Accuracy (IRT Test)	Accuracy (Held out test)
M1	1.16 (87.86)	77.27	87.27
M2	0.461 (67.79)	75.0	86.41

Table 2.6: Estimated ability (θ) and held-out test set accuracy for two LSTM models trained with a full training set (M1) and a sampled training data set (M2). Differences in θ are larger than differences in accuracy and better indicate the gap in model performance.

us to identify a set of examples that when taken together as a test set, can provide a meaningful evaluation of NLP systems with the different difficulty and discriminating characteristics of the examples taken into account. We demonstrate the usefulness of the IRT-generated test set by showing that high accuracy does not necessarily indicate high performance when compared to a population of humans.

IRT is not without its challenges. A large population is required to provide the initial responses in order to have enough data to fit the models; however, crowdsourcing allows for the inexpensive collection of large amounts of data. An alternative methodology is Classical Test Theory, which has its own limitations, in particular that it is test-centric, and cannot provide information for individual examples.

Future work can adapt this analysis to create evaluation mechanisms for other NLP tasks. The expectation is that methods that perform well using a standard accuracy measure can be stratified based on which types of examples they perform well on, and also perform well when the models are used together as an overall test of ability. The hope is that this new method of evaluating NLP systems can lead to new and innovative methods can be tested against a novel benchmark for performance, instead of gradually incrementing on a classification accuracy metric.

CHAPTER 3

UNDERSTANDING DEEP LEARNING MODEL PERFORMANCE THROUGH TEST SET DIFFICULTY

3.1 Introduction

Interpreting the performance of deep learning models beyond test set accuracy is challenging. Characteristics of individual data points are often not considered during evaluation, and each data point is treated equally. In this chapter we examine the impact of a test set question's difficulty to determine if there is a relationship between difficulty and performance. Experiments on Natural Language Inference (NLI) and Sentiment Analysis (SA) show that the likelihood of answering a question correctly is correlated with the question's difficulty. As DNNs are trained with more data, easy examples are learned more quickly than hard examples.

One method for interpreting deep neural networks (DNNs) is to examine model predictions for specific input examples, e.g. testing for shape bias as in [Ritter et al., 2017]. In the traditional classification task, the difficulty of the test set examples is not taken into account. The number of correctly-labeled examples is tallied up and reported. However, it may be worthwhile to use difficulty when evaluating DNNs. For example, what does it mean if a trained model answers the more difficult examples correctly, but cannot correctly classify what are seemingly simple cases? Recent work has shown that for NLP tasks such as Natural Language Inference (NLI), models can achieve strong results by simply using the hypothesis of a premise-hypothesis pair and ignoring the premise entirely [Gururangan et al., 2016, Tsuchiya, 2018, Poliak et al., 2018].

We consider understanding DNNs by looking at the difficulty of specific test set examples and comparing DNN performance under different training scenarios. Do DNN models learn examples of varying difficulty at different rates? If a model does well on hard examples and poor on easy examples, has it has really learned anything? In contrast, if a model does well on easy examples, because a data set is all easy, have the particular task really been "solved"?

As before, methods from Item Response Theory (IRT) are used to model difficulty [Baker and Kim, 2004]. IRT is used to model the difficulty of test examples to determine how DNNs learn examples of varying difficulty. IRT provides a well-studied methodology for modeling example difficulty as opposed to more heuristic-based difficulty estimates such as sentence length. In chapter 2 we used IRT to build new test sets for the NLI and SA tasks and showed that model performance is dependent on test set difficulty. Here the focus is to use IRT to probe specific examples to try to analyze model performance at a more fine-grained level, and expand the analysis to include the task of SA.

We train three DNNs models with varying training set sizes to compare performance on two NLP tasks: NLI and Sentiment Analysis (SA). These experiments show that a DNN model's likelihood of classifying an example correctly is dependent on the example's difficulty. In addition, as the models are trained with more data, the odds of answering easy examples correctly increases at a faster rate than the odds of answering a difficult example correctly. That is, performance starts to look more human, in the sense that humans learn easy examples faster than they learn hard examples.

That the DNNs are better at easy examples than hard examples seems intuitive but is a surprising and interesting result since the example difficulties are modeled *from human data*. There is no underlying reason that the DNNs would find examples that are easy for humans inherently easy. This is the first work to use a grounded measure of difficulty learned from human responses to understand DNN performance.

Premise	Hypothesis	Label	Difficulty
A little girl eating a sucker	A child eating candy	Entailment	-2.74
People were watching the tournament in the stadium	The people are sitting outside on the grass	Contradiction	0.51
Two girls on a bridge dancing with the city skyline in the background	The girls are sisters.	Neutral	-1.92
Nine men wearing tuxedos sing	Nine women wearing dresses sing	Contradiction	0.08

Table 3.1: Examples of sentence pairs from the SNLI data sets, their corresponding gold-standard label, and difficulty parameter (b_i) as measured by IRT (§2.1).

Phrase	Label	Difficulty
The stupidest, most insulting movie of 2002's first quarter.	Negative	-2.46
Still, it gets the job done - a sleepy afternoon rental.	Negative	1.78
An endlessly fascinating, landmark movie that is as bold as anything the cinema has seen in years.	Positive	-2.27
Perhaps no picture ever made has more literally showed that the road to hell is paved with good intentions.	Positive	2.05

Table 3.2: Examples of phrases from the SSTB data set, their corresponding gold-standard label, and difficulty parameter (b_i) as measured by IRT (§2.1).

As deep learning models are trained with larger data sets, the odds of answering easy examples correctly increases at a faster rate than the odds of answering a difficult example correctly. That is, performance starts to look more human, in the sense that humans learn easy things faster than they learn hard things. This result is not as intuitive as it seems, as a deep learning model has no reason to consider examples that are easy for humans as easy.

3.2 Methods

3.2.1 Data

In this chapter the focus is on the difficulty parameter b_i, which represents the midpoint between the upper and lower asymptotes of the item characteristic curve [Baker and Kim, 2004]. Low values of b_i are associated with easier examples (since

an individual with low ability has a 50% chance of answering correctly), and higher values of b_i represent more difficult examples.

To estimate example difficulties for NLI and SA, we used the pre-trained IRT models from the test-set generation (Ch. 2) and extracted the difficulty example parameters. Tables 3.1 and 3.2 show examples of the examples in the data sets, and the difficulty values estimated from the IRT models. The first example in Table 3.1 is a clear case of *entailment*, where if you assume that the premise is true, you can infer that the hypothesis is also true. The label of the second example in SNLI is *contradiction*, but in this case the result is not as clear. There are sports stadiums that offer lawn seating, and therefore this could potentially be a case of entailment (or neutral). Either way, one could argue that the second example here is more difficult than the first. Similarly, the first two examples of Table 3.2 are interesting. Both of these examples are labeled as *negative* examples in the data set. The first example is clear, but the second one is more ambiguous. It could be considered a mild complement, since the author still endorses renting the movie. Therefore you could argue again that the second example is more difficult than the first. The learned difficulty parameters reflect this difference in difficulty in both cases.

3.2.2 Models

For the analyses we trained three representative deep learning models. Each model was trained according to the original parameters provided in the respective papers. Word embeddings for all models were initialized with GloVe 840B 300D word embeddings [Pennington et al., 2014].

3.2.2.1 Long Short Term Memory

The Long Short Term Memory (LSTM) model used here was provided by [Bowman et al., 2015] with the release of the SNLI corpus. The model consists of two LSTM sequence-embedding models [Hochreiter and Schmidhuber, 1997], one to encode the

premise and another to encode the hypothesis. The two sentence encodings are then concatenated and passed through three tanh layers. Finally, the output is passed to a softmax classifier layer to output probabilities over the task classes. For SA, we kept the same architecture but used a single LSTM layer to encode the input text. We implemented this model in DyNet [Neubig et al., 2017].

3.2.2.2 Convolutional Neural Network

We used the convolutional neural network (CNN) model of [Kim, 2014] in our experiments. For each input, the word vector representation of the input tokens were concatenated together to form a matrix. A series of convolutional operations were applied, followed by a max-pooling operation and a fully connected softmax classifier layer. More concretely, for an input sentence $\mathbf{x}$, let $\mathbf{x}_i$ be the word vector representation of the i-th word in $\mathbf{x}$. The convolution operation of filter $\mathbf{w}$ over a window of length h starting with word $\mathbf{x}_i$ results in a context vector c_i:

$$c_i = f(\mathbf{w} \cdot \mathbf{x}_{i:i+h-1} + b) \tag{3.1}$$

where b is a bias term [Kim, 2014]. The filter is applied over all windows in the sentence to generate a feature-map, and max-pooling is used to identify the feature for this particular filter. The process is repeated with multiple filters, and the output features are then passed to a softmax classification layer to output probabilities over the class labels [Kim, 2014]. For NLI, the premise and hypothesis sentences were concatenated before encoding.

3.2.2.3 Neural Semantic Encoder

Neural Semantic Encoder (NSE) is a memory-augmented neural network that uses *read*, *compose*, and *write* operations to evolve and maintain an external memory [Munkhdalai and Yu, 2017] M during training and outputs an encoding h that is used for downstream classification tasks:

$$o_t = f_r^{LSTM}(x_t) \tag{3.2}$$

$$z_t = softmax(o_t^\top M_{t-1}) \tag{3.3}$$

$$m_{r,t} = z_t^\top M_{t-1} \tag{3.4}$$

$$c_t = f_c^{MLP}(o_t, m_{r,t}) \tag{3.5}$$

$$h_t = f_w^{LSTM}(c_t) \tag{3.6}$$

$$M_t = M_{t-1}(\mathbf{1} - (z_t \otimes e_k)^\top) + (h_t \otimes e_l)(z_t \otimes e_k)^\top \tag{3.7}$$

where f_r^{LSTM} is the read function, f_c^{MLP} is the composition function, f_w^{LSTM} is the write function, M_t is the external memory at time t, and $e_l \in R^l$ and $e_k \in R^k$ are vectors of ones [Munkhdalai and Yu, 2017].

For NLI, the premise and hypothesis sentences were each encoded with an NSE module. The outputs were combined and passed through a softmax classifier layer to output probabilities. For SA, we kept the same architecture but used a single NSE layer to encode the input text. We used the publicly available version of the NSE model released by the authors[1] implemented in Chainer [Tokui et al., 2015], and followed the original NSE training parameters and hyperparameters [Munkhdalai and Yu, 2017].

3.2.3 Experiments

The goal in this chapter is to understand how DNN performance on examples of varying difficulty changes under different training scenarios. To test this, we trained three DNN models using subsets of the original SNLI and SSTB training data sets. For each task (NLI and SA), we randomly sampled subsets of training data, from 100 examples up to and including the full training data sets. We sampled 100, 1000,

[1] `https://bitbucket.org/tsendeemts/nse`

2000, 5000, 10000, 50000, 100000, 200000, and 500000 examples for NLI, and sampled 100, 1000, 5000, 10000, 50000, and 75000 examples for SA. We trained each model on the training data subsets, using the original development sets for early stopping to prevent overfitting. The IRT data with difficulty estimates were used as test sets for the trained models. For the IRT data, difficulty parameters were estimated from the Amazon Mechanical Turk response pattern data (Ch. 2). The same test set is used for each of the model/training set size configurations so that we can compare across them.

Once the models were trained and had classified the IRT data sets, we fit logistic regression models to predict whether a DNN model would label an example correctly, using the training set size and example difficulty as the dependent parameters.

3.3 Results

Figure 3.1 plots the contour plots of the learned regression models. The top row plots results for the NLI task, and the bottom row plots results for the SA task. From left to right in both rows, the plots show results for the LSTM, CNN, and NSE models. In each plot, the x-axis is the training set size, the y-axis is the example difficulty, and the contour lines represent the log-odds that the DNN model would classify an example correctly. As the plots show, example difficulty has a clear effect on classification. Easier examples have higher odds of being classified correctly across all of the training set sizes. In addition, the slopes of the contour lines are steeper at lower levels of difficulty. This indicates that, moving left to right along the x-axis, a model's odds of answering an easy example correctly increase more quickly than the odds of answering a harder example correctly.

The contour plots for the CNN and NSE models on the SA task (Figure 3.1, second row middle and right plots) show that the easier examples have higher likelihood of being classified correctly, but the odds for the most difficult examples decrease as

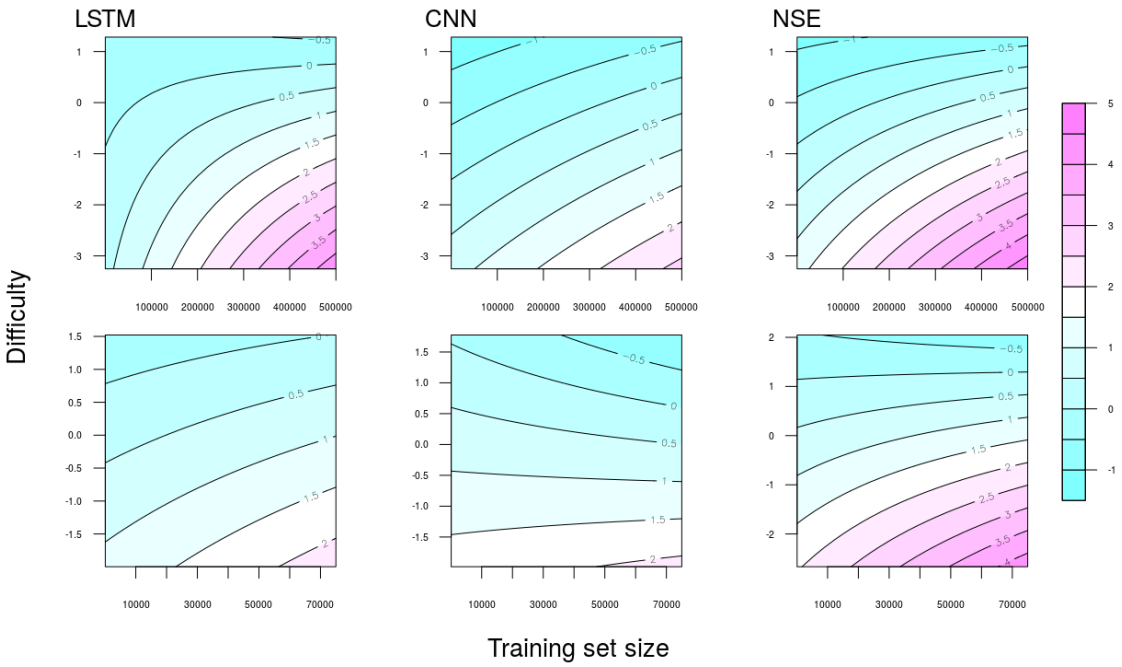


Figure 3.1: Contour plots showing log-odds of labeling an example correctly for NLI (top row) and SA (bottom row) as a function of training set size (x-axis) and example difficulty (y-axis). Each line in the plots represents a single log-odds value for labeling an example correctly. Blue indicates low log-odds of labeling an example correctly, and pink indicates high log-odds of labeling an example correctly. The contour colors are consistent across plots and log-odds values are shown in the legend on the right.

training size increases. This suggests that these models are learning in such a way that improves performance on easy examples but has a negative effect on hard examples. This result is important for interpretability, as it could inform stakeholder decisions if they need to have difficult examples classified.

For the DNN models, there were several subsets of examples with interesting results. Each DNN model (LSTM, CNN, NSE) had at least one subset regression with a negative difficulty coefficient, a positive training set size coefficient, and a negative interaction coefficient. For an example with difficulty 0, as more training data is used, performance increases, which is expected. The negative coefficient of interaction means that the negative slope associated with difficulty is flatter when training size is smaller (e.g. difficult has a smaller effect on performance). In addition, this interaction

parameter tells us that the positive association with training size is steeper for easier examples (examples with difficulty less than 0) and flatter for harder examples. In other words, easier examples are easier to learn than the difficult examples. The slope of performance (in terms of log-odds) with respect to log of training size decreases with example difficulty, indicating that more difficult examples are harder to learn and have a flatter learning curve. This negative interaction also means that the slope of performance with respect to example difficulty decreases with training size. It could be that the training data sets simply have more easy examples in them than difficult examples. In this case, it is important to understand what types of examples are in a particular training set, as it will affect the ability of a model to predict certain types of examples in the future. At the same time, examples may be easy or difficult for different reasons, so deeper analysis of the characteristics of data sets is required.

This behavior, where the expected probability of answering correctly is higher for easy examples than difficult examples, is consistent with the assumptions of IRT models when estimating human ability. A human with a particular estimated ability level will have probability of 0.5 of correctly answering a question with a difficulty parameter equal to his or her ability. This probability increases for easier examples and approaches 1 for the easiest examples. For harder examples, probability decreases and approaches 0.

The idea that easy examples should be easier than hard examples is consistent with learning strategies in humans. For example, when teaching new concepts to students, easier concepts are presented first so that the students can learn patterns and core information before moving to more difficult concepts [Collins et al., 1988, Arroyo et al., 2010]. As students do more examples, all questions get easier, but easy questions get easier at a faster rate. This result is also consistent with the key assumptions of curriculum learning [Bengio et al., 2009].

With more training data, the models move away from treating each question equally regardless of difficulty to a structure more consistent with that of human learning, where the probability of answering an easy question correctly is higher than that of answering a difficult question correctly. This aligns with the expectation that there is a higher probability of answering an easy question than a harder one. As this interaction is evident in each DNN model that was tested, by increasing training size for an NLP model, not only does the expected overall performance increase [Halevy et al., 2009], but the models exhibit a more human-like learning capability with respect to the difficulty of the test set examples.

3.4 Analysis

We conducted additional analysis on the NLI data to see if there were other interesting characteristics associated with these data.

3.4.1 Model Performance

Table 3.3 shows the results of evaluating the models listed above when trained on the original SNLI training set (550k examples). We report ability percentiles on the IRT test sets and accuracy on the SNLI test set as a baseline comparison of models. Recall that for SNLI we obtained 5 different IRT test sets, split according to gold label and the number of quality control annotators that agreed in the original SNLI data set creation. Table 3.3 shows that the RNN models (LSTM and NSE) outperform the CNN model in both accuracy and IRT. However the IRT scores are much more sensitive in terms of identifying high performing RNN models. The NSE model, which had high overall accuracy (84.06%), scores lower in the IRT metrics when comparing with the LSTM model.

Because IRT considers the individual examples that are answered correctly, it may be the case that the specific response pattern for the NSE output is associated with

	Theta Percentile Scores					
Model	5E	5C	5N	4C	4N	Acc
LSTM	**44.83**	**93.82**	**66.28**	**96.25**	**67.00**	77.6
CNN	1.88	47.89	14.63	78.40	28.46	67.0
NSE	14.87	62.76	34.84	90.60	57.79	**84.1**

Table 3.3: Theta Percentile Scores of tested models on the full SNLI training set. Each column refers to one of the 5 SNLI IRT test sets (§2.2.6).

a lower ability score than the LSTM response pattern. The overall accuracy score does not consider the difficulty and discriminatory parameters of individual examples, and therefore cannot distinguish models on the basis of ability. When you consider responses to the IRT test set examples, the LSTM and NSE models answer a similar number of examples correctly (in fact, for some tests the NSE model answers one more question correctly), but the specific examples that are answered correctly affect the ability scores from the IRT models. In addition, there is a larger gap between IRT scores than accuracy scores, which helps when identifying high-performing models and distinguishing between randomness in performance scores.

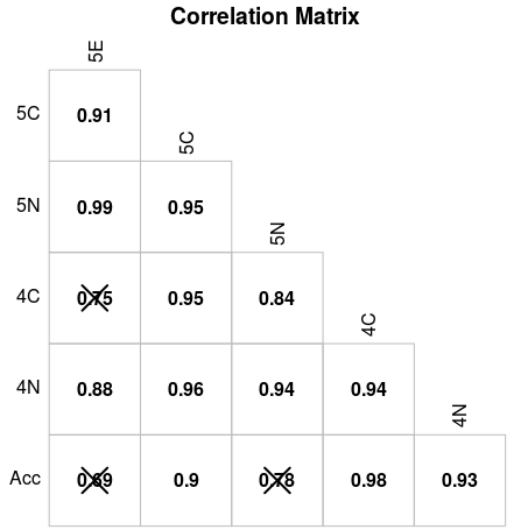


Figure 3.2: Correlation matrix for theta scores and SNLI test set accuracy. Correlations that are not significant ($p < 0.05$) are crossed out.

To confirm the validity of IRT as a metric the correlations between accuracy and the IRT scores are reported to see how consistent the scores are (Figure 3.2). The IRT percentile scores are highly correlated with accuracy (above 0.8 for 5C, 4C, and 4N, 0.69 for 5N and 0.59 for 5E). In addition, each IRT score is highly correlated with each other (≥ 0.80 for all pairs except 4C and 5E, which is 0.73). All but 3 correlations are statistically significant ($p < 0.05$). The strong positive correlation between IRT scores and accuracy shows that IRT as an evaluation metric is consistent with existing metrics.

3.5 Discussion

We have shown that DNN model performance is affected by example difficulty as well as training set size. This is the first work that has used a well-established method for estimating difficulty to analyze DNN model performance as opposed to heuristics. DNN models perform better on easy examples, and as more data is introduced in training, easy examples are learned more quickly than hard examples. Learning easy examples faster than harder examples is what would be expected when examining human response patterns as they learn more about a subject. However this has not previously been shown to be true in DNN models. As more training examples are used for training, mastery of easy examples comes more quickly than of harder examples. Figure 3.1 plots the predicted log-odds of answering an example correctly based on the regression models. In each case, the odds increase as the training set size increases, which is to be expected. However, the odds increase more quickly for easier examples than for harder examples.

That the results are consistent across NLI and SA shows that the methods can be applied to a number of NLP tasks. The SA results do show that the odds of labeling a difficult example correctly decrease with more training data 3.1. It could be the case that these difficult examples in the SA task are more subjective than the easier

examples, for example a review that is fairly neutral and is split between positive and negative annotations. These cases would be more difficult for a model to label, and are worth examining in more detail. By identifying examples such as these as difficult makes it easier to see where the model is going wrong and allows for research on better way to represent these cases.

This result has implications for how machine learning models are evaluated across tasks. The traditional assumption that the test data is drawn from the same distribution as the training data, makes it difficult to understand how a model will perform in settings where that assumption does not hold. However, if the difficulty of test set data is known, one can better understand what kind of examples a given model performs well on, and specific instances where a model underperforms (e.g. the most difficult examples). In addition, researchers can build test sets that consist of a specific type of data (very easy, very hard, or a mix) to evaluate a trained model under specific assumptions to test generalization ability in a controlled way. This could allow for more confidence in model performance in more varied deployment settings, since there would be a set of tests a model would have to pass before being deployed.

It is important to note that the difficulty parameters were estimated *from a human population*, meaning that those examples that are difficult for humans are in fact more difficult for the DNN models as well. This does not need to be the case given that DNNs learn very different patterns, etc. than humans. In fact there were exceptions in the results which shows that these models should be carefully examined using techniques like those described here. Future work can investigate why this is the case and how this information can be leveraged to improve model performance and interpretability.

Evaluating progress in Machine Learning requires effective metrics to measure algorithms output on test sets. We demonstrate the reliability of IRT as a metric for NLI. IRT scores are consistent in that they separate models based on performance

in a similar manner as accuracy, while providing more information with regards to the examples answered correctly by each model. These experiments have shown that the IRT metric is consistent with the standard accuracy results in identifying high-performing and low-performing models. A key benefit of IRT is that provides a way to estimate parameters of individual examples, such as difficulty. Estimating example parameters allows for more fine-grained analysis of model performance. The results show that as DNN models are trained with larger data sets, their performance begins to look like performance expected from humans. That is, the probability that the model will answer an easy question correctly is much higher than the probability that it will answer a more difficult question correctly. These difficulty parameters are modeled on human response data, which makes the result all the more interesting. Those questions that humans find hard are also hard for the models. As the DNN models are trained with more data, learning patterns emerge that mirror those of humans. Whereas with little training, easy and difficult examples have similar likelihood of being answered correctly, with larger training sizes easy examples have a higher likelihood of being answered correctly.

These results use parameters learned from human response patterns when estimating ability. IRT makes it possible to estimate a model's ability with regard to the original human population. This allows us to place a model on a scale of ability that is directly comparable to humans. A traditional metric like accuracy is dependent on the data set. If a data set is very easy, high accuracy scores are not necessarily indicative of high performance. This is reflected in scores for the 5E IRT test set, which is very easy. Conversely, a test set is hard, then low accuracy does not imply low ability (e.g. the 4C test set).

These results are dependent on the difficulties estimated from a human population of AMT annotators. Therefore it is possible that certain subsets of questions had a greater influence on the IRT models than others. A larger set of examples in the

IRT models could reduce the implicit weighting of certain questions and have a more appropriate distribution of ability levels. This is difficult due to the need for human annotators, but automating response pattern generation would be an interesting direction for future work.

Future work can explore additional models using more specialized features to attempt to improve performance. Ensemble models that consider the output of multiple DNNs (e.g. CNNs, RNNs, and memory networks) can take advantage of the high performance of different categories of sentence pair to further improve performance.

CHAPTER 4

SOFT-LABEL MEMORIZATION-GENERALIZATION

4.1 Introduction

Often when multiple labels are obtained for a training example it is assumed that there is an element of noise that must be accounted for. It has been shown that this disagreement can be considered signal instead of noise [Inel and Aroyo, 2017]. In this chapter we investigate using soft labels for training data to improve generalization in machine learning models. However, using soft labels for training Deep Neural Networks (DNNs) is not practical due to the costs involved in obtaining multiple labels for large data sets. We propose soft label memorization-generalization (SLMG), a fine-tuning approach to using soft labels for training DNNs. We assume that differences in labels provided by human annotators represent ambiguity about the true label instead of noise. Experiments with SLMG demonstrate improved generalization performance on the natural language inference (NLI) and sentiment analysis (SA) tasks. By injecting a small percentage of soft label training data (0.03% of training set size) we can improve generalization performance over several baselines.

In Machine Learning (ML) classification tasks a model is trained on a set of labeled data and optimized based on some loss function. The training data consists of some feature set $X_{\text{train}} = x_i, \ldots, x_N$ and associated labels $Y_{\text{train}} = y_1, \ldots, y_N$, where Y is a vector of integers corresponding to the classes of the problem. For binary classification, Y would be a vector of 0's and 1's, with 0 representing the negative class and 1 representing the positive class. The goal when training an ML classification model is to minimize the error in the model's prediction of a class label for a given training

example. The loss function can take many forms, but at a high level the goal is to minimize the number of training examples the model misclassifies:

$$\sum_{i}^{N} \mathbb{I}[\hat{y}_i \neq y_i] \tag{4.1}$$

where $\mathbb{I}[x]$ is the indicator function. It is typically assumed that each training example is labeled correctly, and each is equally appropriate for a single class. There is no way to quantify the uncertainty of the examples, nor a way to exploit such uncertainty during training. Particularly for NLP tasks with sentence- or phrase-based classification such as natural language inference (NLI) and sentiment analysis (SA), it is not common to model ambiguity in language in training data labels.

For example, consider the following two premise-hypothesis pairs, both taken from the Stanford Natural Language Inference (SNLI) corpus for NLI [Bowman et al., 2015]:

1. *Premise:* Two men and a woman are inspecting the front tire of a bicycle.
   Hypothesis: There are a group of people near a bike.

2. *Premise:* A young boy in a beige jacket laughs as he reaches for a teal balloon.
   Hypothesis: The boy plays with the balloon.

In both cases the gold-standard label in the SNLI data set is *entailment*, which is to say that if you assume that the premise is true, you can infer that the hypothesis is also true. However, looking at the two sentence pairs one could argue that they do not both equally describe entailment. The first example is a clear case: people inspecting a front tire of a bike are almost certainly standing near it. However the second example is less clear. Is the child laughing because he is playing? Or is he laughing for some other reason, and is simply grabbing for the balloon to hold it (or give it to someone else)? One could argue that a laughing child is more often than not associated with play, but that requires additional external knowledge that might not

be contained in the data set. There is ambiguity associated with the two examples that is not captured in the data. To a machine learning model trained on SNLI, both examples are to be classified as entailment, and incorrect classifications should be penalized equally during learning.

Previous work has shown that leveraging crowd disagreements can improve the performance of named entity recognition (NER) models by treating disagreement not as noise but as signal [Inel and Aroyo, 2017]. We use the same assumption here and encode crowd disagreements directly into the model training data in the form of a distribution over labels ("soft labels"). These soft labels model *uncertainty* in training by representing human *ambiguity* in the class labels. Ideally we would have soft labels for all of our training data, however when training large deep learning models it is prohibitively expensive to collect many annotations for all data in the huge data sets required for training. When training deep neural networks (DNNs), even a small amount of soft labeled data can improve generalization.

With this in mind we propose soft label memorization-generalization (SLMG), a fine-tuning approach to training that uses distributions over labels for a subset of data as a supplemental training set for a learning model. Ideally a model could be trained with soft labels for all training examples, but because of the costs involved, only a small number of examples for fine-tuning augment a larger data set.

Our hypothesis is that using labels that incorporate language ambiguity can improve model generalization in terms of test set accuracy, even for a small subset of the training data. By using a distribution over labels we hope to reduce overfitting by not pushing probabilities to 1 for examples where the empirical distribution is more spread out. Results show that SLMG is a simple and effective way to improve generalization without a lot of additional data for training.

We evaluate our approach on NLI using the SNLI data set [Bowman et al., 2015] and SA using the Stanford Sentiment Treebank (SSTB) [Socher et al., 2013]. Prior

work has shown that lexical phenomena in the SNLI data set can be exploited by classifiers without learning the task, and performance on difficult examples in the data set is still relatively poor, making NLI a still-open problem [Gururangan et al., 2016,Poliak et al., 2018]. For soft labeled data we use the IRT evaluation scales for NLI and SA data introduced earlier where each example was labeled by 1000 AMT workers (Chapter 2). This way we are able to leverage an existing source of soft labeled data without additional annotation costs. We find that SLMG can improve generalization under certain circumstances, even thought the amount of soft labeled data used is tiny compared to the total training sets (e.g. 0.03% of the SNLI training data set). SLMG outperforms the obvious but strong baseline of simply gathering more unseen data for labeling and training. Our results suggest that there are diminishing returns for simply adding more data past a certain point [Halevy et al., 2009], and indicate that representing data uncertainty in the form of soft labels can have a positive impact on model generalization.

This chapter presents the following contributions: (i) we propose the SLMG framework for incorporating soft labels in machine learning training, (ii) we use previously-collected human annotated data to estimate soft label distributions for NLI and SA and show that replacing less than 0.1% of training data with soft labeled data can improve generalization for three DNN models, and (iii) we demonstrate for the first time that soft labels can encode ambiguity in training data that can improve model generalization in terms of test set accuracy.

4.2 Soft Label Memorization-Generalization

4.2.1 Overview

In a traditional supervised learning single-label classification problem, a model is trained on some data set X_{train}, and tested on some test set X_{test}. In this setting, learning is done by minimizing some loss function L. We assume that the labels

associated with instances in X_{train} are correct. That is, for each $(x_i, y_i) \in X_{\text{train}}$ we assume that y_i is the *correct* class for the i-th example, where x_i is some set of features associated with the i-th training example and y_i is the corresponding class. However it is often the case, particularly in NLP, that examples may vary in terms of difficulty, ambiguity, and other characteristics that are often not captured by the single correct class to which the example belongs. The traditional single-label classification task does not take this into account.

For example, a popular loss function for classification tasks is Categorical Cross-Entropy (CCE). For a single training example x_i with class $y_i \in Y$ where Y is the set of possible classes, CCE loss is defined as

$$L_i^{\text{CCE}} = -\sum_{j=1}^{|Y|} p(y_{ij}) \log p(\hat{y}_{ij}) \tag{4.2}$$

In the single-class classification case where a single class j has probability 1 CCE loss is

$$L^{\text{CCE}} = \sum_{i}^{N} -\log p(\hat{y}_{ij}) \tag{4.3}$$

where each example loss is summed over all of the training examples. With this loss function a learning model is encouraged to update its parameters in order to maximize the probability of the correct class for each training example. Without some stopping criteria, parameter updates will continue for a given example until $p(\hat{y}_{ij}) = 1$. This may not always be ideal, since by pushing the model output probability to 1, the learner is encouraged to overfit on an example that may not be representative of the particular class.

With SLMG we want to take advantage of the fact that differences between examples in the same class can be useful during training. Instead of treating each training example as having a single correct class, SLMG uses a distribution over labels

for the gold standard. This way examples with varying degrees of uncertainty are reflected during training.

We make a different assumption regarding noise in human generated labels than previous work [Dawid and Skene, 1979, Bachrach et al., 2012]. The presence of noise when multiple labels are obtained is often attributed to labeler error, lack of expertise, adversarial actions, or other negative causes. However, we believe that the noise in the labels can be considered a *signal* [Inel et al., 2014, Aroyo and Welty, 2015]. Examples with less uncertainty about the label (in the form of a label distribution with a single high peak) should be associated with similarly high model confidence.

4.2.2 Learning with SLMG

In our experiments we investigated two ways to incorporate the soft labeled data into model training, which we define below. Let X_{train} be the training set with one-hot gold labels, and let X_{test} be the test set. Let X_{soft} be the soft labeled training data with class probabilities. We assume that there is no overlap between the examples in X_{train} and X_{soft}: $X_{\text{train}} \cap X_{\text{train}} = \emptyset$. There are two ways to incorporate the X_{soft} data into a learning task that we investigate: (i) at each training epoch, training with X_{train} and X_{soft} *interspersed* (SLMG-I), and (ii) train a model on X_{train} for a predefined number of epochs, followed by training on X_{soft} for a predefined number of epochs, repeated some number of times (*meta-epochs*) in a *sequential* fashion (SLMG-S). Algorithms 1 and 2 define the two training sequences, respectively. In our experiments we tested two loss functions for the SLMG data, CCE (§4.2.1) and Mean Squared Error (MSE): $L_i^{\text{MSE}} = \sum_{j=1}^{|Y|}(\hat{p}(y_{ij}) - p(y_{ij}))^2$.

4.2.2.1 Interspersed Fine-Tuning

The motivation for interspersing fine-tuning with soft labels is to prevent overfitting as the model learns. After each epoch in the training cycle, the learning model will have made updates to the model weights according to the outputs on the full training

Algorithm 1 SLMG-I Algorithm

Input: Model m, NumEpochs e, X_{train}, X_{soft}
for $i = 1$ **to** e **do**
    Train m on X^N_{train}
    Train m on X_{soft}
end for

set. By interspersing the fine-tuning after each epoch, using soft labels can account for and correct overfitting earlier in the process by making smaller updates to the model weights according to the soft label distributions. This method encourages generalization early in the process, before the model can memorize the training data and possibly overfit.

4.2.2.2 Sequential Fine-Tuning

In contrast with the interspersed fine-tuning, the motivation for sequential fine-tuning is to adjust a well-trained model to improve generalization. After a full training cycle of some number of epochs, the learning model is then fine-tuned using the soft-labeled data. This way the fine-tuning takes place after the model has learned a set of weights that perform well on the training data. Fine-tuning here can improve generalization by updating the model weights to be less extreme when dealing with examples that are more ambiguous than others. Since these updates happen on a trained model, there is less risk of the model performance drastically reducing. By repeating this process over a number of meta-epochs, the learning model can memorize, generalize, and repeat the cycle.

4.2.3 Collecting Soft Labeled Data

For soft labeled data, we use the AMT response pattern data collected for earlier work (Chapter 2). 180 SNLI training examples split evenly between the three labels were randomly selected and given to Amazon Mechanical Turk (AMT) workers (Turkers) for additional labeling. For each example 1000 additional labels were

Algorithm 2 SLMG-S Algorithm

Input: Model m, NumMetaEpochs me, NumEpochs e, X_{train}, X_{soft}
for $i = 1$ **to** me **do**
  for $j = 1$ **to** e **do**
    Train m on X^N_{train}
  end for
  for $j = 1$ **to** e **do**
    Train m on X_{soft}
  end for
end for

collected. In order to estimate a distribution over labels for these examples we calculate the probability of a certain label according to the proportion of humans that selected the label: $P(Y = y) = \frac{N_y}{N}$, where N_y is the number of times y was selected by the crowd and N is the total number of responses obtained.

For SA, 134 examples were randomly selected from SSTB, using a similar AMT setup as for the SNLI data. For each randomly selected example, we asked 1000 Turkers to label the sentence as very negative, negative, neutral, positive, or very positive.

Table 4.1 shows example premise-hypothesis pairs taken from the SNLI data set for NLI [Bowman et al., 2015]. Table 4.1 includes the premise and hypothesis sentences, the gold standard class as included in the data set, as well as estimated soft labels using human responses.[1] There are premise-hypothesis pairs that share a class label (e.g. the first two examples) yet are very different in terms of how they are perceived by a crowd of human labelers. In a traditional setup both examples would have a single class label associated with contradiction (class label 1 if 0 = *entailment*, 1 = *contradiction*, and 2 = *neutral*). Certain training examples have much less uncertainty associated with them, which is reflected in the high probability weight on the correct label. In other cases, there is a more evenly spread distribution, which can be interpreted as

[1]Typos in the examples are from the original data set and are preserved intentionally.

a higher degree of uncertainty. In a learning scenario, one may want to treat these examples differently according to their uncertainty, as opposed to the common practice of weighing each equally. Similarly Table 4.2 shows examples of sentences for the SA task that demonstrate how the gold standard label provided in the data set does not capture the uncertainty with the labels provided by the crowd.

Premise	**Hypothesis**	P(**E**)	P(**C**)	P(**N**)
A little boy is opening gifts surrounded by a group of children and adults.	The boy is being punished	0.005	**0.839**	0.156
A man and woman walking away from a crowded street fair.	There are a group of men walking together.	0.045	**0.542**	0.412
Two men and a woman are inspecting the front tire of a bicycle.	There are a group of people near a bike.	**0.861**	0.032	0.108
A young boy in a beige jacket laughs as he reaches for a teal balloon.	The boy plays with the balloon.	**0.659**	0.026	0.316
A man wearing a gray shirt waving in the middle of a plant nursery	The man does not have a way to get home.	0.011	0.174	**0.815**
A wielder works on wielding a beam into place while other workers set beams.	The wielder is working on a building.	0.486	0.013	**0.501**

Table 4.1: Examples of premise-hypothesis pairs from the SNLI data set and the AMT-estimated probability that the correct label is Entailment (**E**), Contradiction (**C**), or Neutral (**N**). The original gold-standard label from SNLI is in bold. In some cases, the gold label provided originally has a low probability based on AMT-population estimates (i.e. less than 75%).

Sentence	P(**VNeg**)	P(**Neg**)	P(**Neu**)	P(**Pos**)	P(**VPos**)
An endlessly fascinating, landmark movie that is as bold as anything the cinema has seen in years.	0.01	0.015	0.023	0.128	**0.824**
If no one singles out any of these performances as award-worthy, it's only because we would expect nothing less from this bunch.	0.061	0.148	0.164	0.297	**0.329**
Trivial where it should be profound, and hyper-cliched where it should be sincere	**0.421**	0.416	0.093	0.048	0.021

Table 4.2: Examples from the SSTB data set and the AMT-estimated probabilities over labels. The gold label from SSTB is in bold.

Consider calculating the entropy, $H(X)$, of the first two training examples from Table 4.1:

$$H(X) = -\sum_{y \in Y} p(y) \log p(y) \tag{4.4}$$

If we assume that the probability of the correct label (in this case, contradiction), is 1, and the probability of all other labels is 0, then entropy in both cases is 0.[2] However if we use the distributions from Table 4.1, then entropy is 0.464 and 0.837 respectively. There is much more uncertainty in the second example than the first, which is not reflected if we assume that both examples are labeled contradiction with probability 1. This uncertainty may be important when learning for classification.

4.2.4 Learning from the Crowd

In this chapter we take advantage of the fact that we have a distribution over labels provided by the human labelers. We can train using CCE or MSE as our loss function, where we minimize the difference between the estimated probabilities learned by the model and the empirical distributions obtained from AMT over the training examples. SLMG attempts to move the model predictions closer to the soft label distribution of responses. SLMG is not necessarily trying to push predicted probability values to 1, which is a departure from the typical understanding of single label classification in ML. Here I hypothesize that updating weights according to differences in the observed probability distributions will improve the model by preventing it from updating too much for more uncertain examples (that is, examples where the empirical distribution is more evenly spread across the three labels).

This scenario assumes that the crowdsourced distribution of responses is a better measure of correctness than a single gold-standard label. We hypothesize that the crowd distribution over labels gives a fuller understanding of the examples being used for training. SLMG can update parameters to move closer to this distribution without making large parameter updates under the assumption that a single correct label should have probability 1.

---

[2]Where $0 \log 0 = 0$.

If we assume that ML performance is not at the level of an average human (which is reasonable in many cases), then SLMG can help pull models towards average human behavior when we use human annotations to generate the soft labels. If the model updates parameters to minimize the difference between predictions and the distribution of responses provided by AMT workers, then the model predictions should look like that of the crowd. When ML model performance is better than the average AMT user, there is a risk that performance may suffer, if we assume that our model would outperform a human population. The model may have learned a set of parameters that better models the data than the human population, and updating parameters to reflect the human distribution could lead to a drop in performance. However since SLMG is only used as a fine-tuning mechanism, the risk here is mitigated by the larger training set used alongside the SLMG data.

4.3 Experiments

Our hypothesis is that soft labeled data, even in very small amounts, can improve model generalization by capturing ambiguity of language data in the form of distributions over labels. We describe our experiments to test this hypothesis, as well as the data sets and models used in the experiments. At a high level, our goal is to understand how distributions over labels can affect the learning process. To do this we look at several ways of incorporating the SLMG data. By varying the point at which we inject the SLMG data we can observe how performance is affected.

For the experiments in this chapter we utilize the same data sets (§2.2.1) as studied earlier. For deep learning models, we use both the LSTM and NSE models described earlier (§3.2.2) as well as the Enhanced Sequential Inference Model (ESIM) [Chen et al., 2017]. ESIM consists of three stages: (i) input premise and hypothesis encoding with BiLSTMs, (ii) local inference modeling with attention, and (iii) inference composition with a second BiLSTM encoding over the local inference information.

We used the publicly available ESIM model released by the authors[3] implemented in Theano [Theano Development Team, 2016] and kept all of the hyperparameters the same as in the original paper. We do not use ESIM in our SA experiments. The model was designed specifically for NLI, as opposed to NSE which performs well across several tasks including sentiment analysis [Munkhdalai and Yu, 2017]. In addition, NSE performance on SA is close to state-of-the-art, so testing another high-performing model in this case is unnecessary.

The SA soft label data examples were selected from the SSTB test set, so for our experiments we use a modified SSTB test set where the examples have been removed. In our results we report baseline scores on the modified test set so as to be consistent. We chose to select from the SSTB test set because the training set for SSTB, particularly for the binary task, is smaller than the SNLI data set. We would rather keep all data for training in this instance, and report all of our results on a smaller, but still substantial test set. For all experiments we used early stopping and report test results for the epoch with the highest dev set performance.

4.3.1 Baselines

We evaluate SLMG against three baselines: (i) *B1, Traditional:* we train the DNN models (§3.2.2) in a traditional supervised learning setup, where the soft labeled training data (X_{soft}) is incorporated in the hard labeled training data (X_{train}) with their original gold-standard labels, (ii) *B2, Comparable Label Effort (CLE):* Because each of the 180 X_{soft} examples have 1000 human annotations, our second baseline is to add new single label training data to B1, to evaluate against a comparable data labeling effort. To that end, we randomly selected 180,000 additional training data points from the Multi-NLI data set [Williams et al., 2018] for additional training data, (iii) *B3, AOC:* The third baseline is the All in one Classifier (AOC) approach [Kajino

[3] `https://github.com/lukecq1231/nli`

et al., 2012], where for each example in X_{soft}, every label obtained from the crowd is used as a unique example in the training data. This baseline also has an addition 180,000 training data points as in B2, but the additional pairs all come from X_{soft} and have varying labels depending on the crowd responses.

4.4 Results and Analysis

Table 4.3 reports results on the SNLI test set. For each model on the NLI task, SLMG leads to improved generalization performance (i.e. test set accuracy) by injecting soft labeled data at some point. Note that the best performance with SLMG varies according to the model, but for each model there is some configuration that does improve performance. As with all model training, the effect of SLMG requires experimentation according to the use case. In all cases, using CCE as the loss function performs better than using MSE. We suspect that this is due to the fact that small differences are penalized less with CCE than with MSE.

Task	**Model**	**Baselines**			**SLMG-S**		**SLMG-I**	
		B1	B2	B3	MSE	CCE	MSE	CCE
NLI	LSTM	76.7	76.9	75.7	76.5	**77.4**	76.9	76.7
	NSE	84.6	84.8	84.0	84.1	**85.1**	84.3	84.4
	ESIM	87.7	84.0	84.1	87.7	87.6	87.8	**87.9**
SA-B	LSTM	87.4	86.7	87.0	87.3	**87.5**	86.5	**87.5**
	NSE	88.9	88.7	87.7	87.5	88.6	88.4	**89.1**
SA-FG	LSTM	49.7	n/a	50.1	50.8	47.0	**51.7**	49.9
	NSE	**52.3**	n/a	50.6	51.0	51.9	52.0	51.2

Table 4.3: Training and test accuracy results for incorporating SLMG in three tasks: NLI, binary sentiment analysis (SA-B), and fine-grain sentiment analysis (SA-FG). Note: for B2, we cannot run on the fine-grained sentiment analysis task because the supplemental data set only includes binary sentiment labels (positive/negative).

For the SA task, injecting SLMG data at some point again improves performance. SLMG does not improve performance for the NSE model on the fine-grained SA task, but for the binary task there are improvements for both the LSTM and NSE models. This suggests that data close to the decision boundary that was originally misclassified

was classified correctly when soft labeled data was added (see Table 4.4 for examples). With binary SA, there is no distinction between "very negative" and "negative" so changes in degree don't have an effect, unless the change is from negative to positive.

Table 4.4 shows examples of premise-hypothesis pairs from the SNLI test set, and the model output probabilities from the B1 baseline and the SLMG-I model trained with CCE as the soft label loss function. In the first three examples, using SLMG results in flipping the output from incorrect to correct. For the first pair, this pair seems to be a weak case of entailment, and could be argued to be neutral. The SLMG model considers this and has a reasonably high probability for the neutral class. In the last example, training with SLMG results in the wrong label, but again it could be argued that this is a case where neutral is appropriate. The "sedan" that is stuck may not be the Land Rover (Land Rovers are SUVs), so neutral is a reasonable output here.

Premise	**Hypothesis**	**Model**	$P(\mathbf{E})$	$P(\mathbf{C})$	$P(\mathbf{N})$
This church choir sings to the masses as they sing joyous songs from the book at a church.	The church is filled with song	B1	**0.191**	0.021	0.788
		SLMG-I-CCE	**0.520**	0.028	0.452
Two women are observing something together.	Two women are looking at a flower together.	B1	0.530	0.066	**0.404**
		SLMG-I-CCE	0.209	0.0270	**0.764**
A older man in a hat is playing a accordion on the street while sitting in a chair.	A man is playing guitar.	B1	0.814	**0.090**	0.096
		SLMG-I-CCE	0.055	**0.827**	0.118
A land rover is being driven across a river.	A sedan is stuck in the middle of a river.	B1	0.014	**0.561**	0.435
		SLMG-I-CCE	0.011	**0.241**	0.749

Table 4.4: Examples of premise-hypothesis pairs from the SNLI data set and output probabilities from the LSTM model. For both examples the probabilities associated with the gold label are in bold.

4.4.1 Changes in Outputs from SLMG

To better understand the effects of SLMG on generalization, we look at the changes in test set performance when SLMG is used as compared to the baseline case. Table 4.5 shows 3 confusion matrices: the test-set output for the baseline LSTM model on the NLI task, and the same model when trained with SLMG-S and CCE as the loss function for the soft labeled data, which improved test set performance and SLMG-S with MSE as the loss function for the soft labeled data, which did not. In both cases of training with SLMG, the number of correctly classified entailment and contradiction examples increased, while the number of neutral examples correctly classified decreased. However when MSE is used as the soft label loss function, the increase in misclassified neutral examples was enough to offset the gains in correctly classified entailment and contradiction examples. Depending on the use case, this result could be useful for applications. Fewer false negatives for entailment and contradiction examples may be more important than fewer true positives for the neutral class.

		E	C	N
Baseline	E	2739	191	438
	C	333	2360	544
	N	441	332	**2446**
SLMG-S (CCE)	E	2828	157	383
	C	375	2401	461
	N	520	328	2371
SLMG-S (MSE)	E	**2967**	158	243
	C	466	**2415**	356
	N	677	422	2120

Table 4.5: Confusion matrices for the LSTM model, trained according to the baseline (first block), using SLMG-S with CCE (second block), and using SLMG-S with MSE (third block). Gold standard labels run down the left hand side, while predicted labels are across the top in the matrix. The highest count of True Positives for each label across the three model-training setups are in bold.

If SNLI is considered as a binary classification task, with two possible labels "entailment" and "not entailment" (where we combine contradiction and neutral), and look at Table 4.5 SLMG outperforms the baseline in both cases. In fact, the

SLMG-MSE method outperforms SLMG-CCE in the binary task (88.0% vs. 86.6%) due to the fact that its performance on the entailment label is much higher.

4.4.2 Comparing the Crowd to the Gold Standard

We also looked at the soft labeled data itself to understand how well the crowd label distributions align with the accepted gold-standard labels in the original data set. Figure 4.1 reports on how well the crowd distributions align with the gold standard labels included in the original data sets (SNLI and SSTB). There are quite a few examples where the gold standard class label does not have a high degree of probability weight as estimated from the crowd. In particular, for fine-grained sentiment classification, the distribution is similar to a normal distribution, with a very small number of examples where the probability associated with the gold standard label is high.

For NLI, there is a high percentage of examples where the gold label has an estimated probability of less than 80%. This may be due to the fact that individuals have different understanding of what constitutes entailment. This uncertainty among humans is useful for understanding outputs from ML models. This is consistent with the inter-rater reliability (IRR) scores reported during our AMT data collection (Chapter 2). Recall that IRR scores (Fleiss' κ) for the data ranged from 0.37 to 0.63, which is considered moderate agreement [Landis and Koch, 1977]. The moderate agreement indicates that there is a general consensus about which label is correct (which is consistent with Figure 4.1), but there is enough disagreement among the annotators that the disagreements should be incorporated into the training data, and not discarded in favor of majority vote or another single label selection criteria.

4.4.3 How Many Labels do you Need?

Of course, collecting 1000 labels per example to estimate soft labels becomes prohibitively expensive very quickly. However it may not be necessary to collect

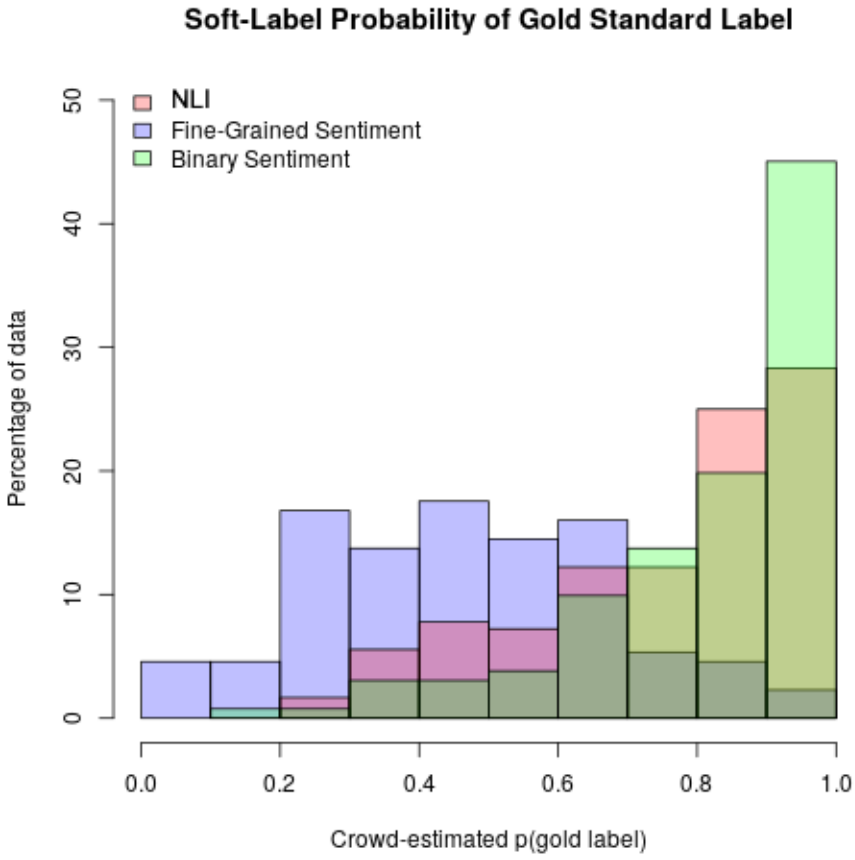


Figure 4.1: Relative frequency histograms for the crowd-estimated probability of the original gold-standard label.

that many labels in practice. To determine how many labels are needed to arrive at a reasonable estimate of the soft label distributions, we randomly sampled crowd workers from our data set one at a time. At each step, we used the sampled workers responses to estimate the soft labels for each example and calculated the Kullback Liebler divergence (KL-Divergence) between the true soft label distributions and the sampled soft label distributions:$D_{KL}(p||q) = -\sum_i P(i) \log \frac{P(i)}{Q(i)}$, where P is the true soft label distribution estimated from the full data set and Q is the sampled soft label distribution. Figure 4.2 plots the KL-Divergence averaged over the number of data

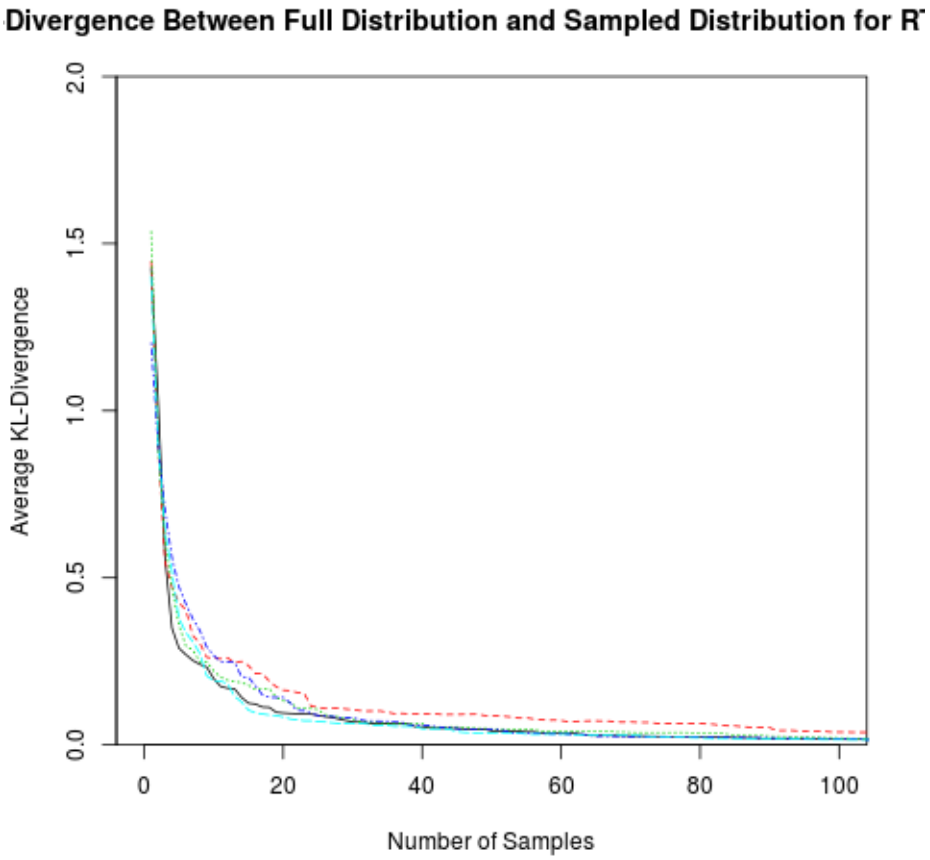


Figure 4.2: Average KL-Divergence between sub-sampled crowd distributions and the estimated soft label distribution from the entire crowd data. Sampling 20 crowd workers achieves a good estimate of the label distributions without the cost of using the full 1000 worker population.

set examples (180) as a function of the number of crowd workers selected.[4] We plot results for 5 runs of the random sampling procedure. As the figure shows, the average KL-Divergence approaches 0 well before all 1000 labels are necessary.

When sampling randomly, the average difference drops very quickly, and is very low with as few as 15 or 20 labels per example. Active learning techniques could reduce this number further, either by selecting "good annotators" or identifying examples for which more labels are needed. This is left for future work.

[4]We truncate the x-axis to focus on the lower values.

To confirm the observation that significantly fewer labels are necessary, we randomly sampled 20 annotators from the data set, used their responses to estimate the soft label distributions, and re-trained the LSTM model with SLMG-I using CCE as the soft label loss function. We ran this training 10 times, where each time we sampled a new selection of 20 annotators for estimating the soft label distributions. The average accuracy for these models was 76.9 and the standard deviation was 0.3. These models perform as well as the model using the distributions learned from 1000 annotators, with significantly less annotation cost.

4.5 Discussion

We have introduced SLMG, a fine-tuning approach to training that can improve classification performance by leveraging uncertainty in data. In the NLI task, incorporating the more informative class distribution labels leads to improved performance under certain training setups. By introducing specialized supplemental data the model is able to update its representations to boost performance. With SLMG, a learning model can update parameters according to a gold-standard that allows for uncertainty in predictions, as opposed to the classic case where each training example should be equally important during parameter updates. Training examples with higher degrees of uncertainty within a human population have less of an effect on gradient updates than those examples where confidence in the label is very high as measured by the crowd.

SLMG is an easy fix, but it is not a silver bullet for improving generalization. In our experiments we found that under different training settings SLMG can improve performance for the different models. It is worthwhile to experiment with SLMG to see if and how it can improve performance on other NLP tasks. NLI is a particularly good use case for SLMG because of the ambiguity inherent in language and the potential disagreements that can arise from different interpretations of text. In

addition, further experimentation with the way soft labels are generated can lead to further generalization improvements.

There are limitations to this work. One bottleneck is the requirement for having a large amount of human labels for a small number of examples, which goes against the traditional strategy for crowdsourcing label-generation. However one can probably estimate a reasonable distribution over labels with significantly fewer labels than obtained here for each example (Figure 4.2). On the other hand, the new SA data set of human response can be used for modeling IRT parameters such as difficulty. Identifying a suitable number using active learning techniques is left for future work.

While SLMG requires soft labels, it does not necessarily require *human-annotated* soft labels. Rather, SLMG only requires some measure of uncertainty between training examples as part of the generalization step. This can come from human annotators, an ensemble of machine learning models, or some other pre-defined uncertainty metric. In our experiments we demonstrate the validity of SLMG using an existing data set from which we can extract soft labels, and leave experimentation with different soft label generation methods to future work.

Future work includes investigation into data sets that can be used with SLMG and why certain fine-tuning sets lead to better performance in certain scenarios. Experiments with different loss functions (e.g. KL-Divergence) and different data can help to understand how SLMG affects the representations learned by a model. Our results suggest that future work training DNNs to learn a distribution over labels can lead to further improvements.

CHAPTER 5

LEARNING LATENT PARAMETERS WITHOUT HUMAN RESPONSE PATTERNS: ITEM RESPONSE THEORY WITH ARTIFICIAL CROWDS

5.1 Introduction

Incorporating Item Response Theory (IRT) into NLP tasks can provide valuable information about model performance and behavior. Traditionally, IRT models are learned using human response pattern (RP) data, presenting a significant bottleneck for large data sets like those required for training deep neural networks (DNNs). In this work we propose learning IRT models using RPs generated from artificial crowds of DNN models. We demonstrate the effectiveness of learning IRT models using DNN-generated data through quantitative and qualitative analyses for two NLP tasks. Parameters learned from human and machine RPs for natural language inference and sentiment analysis exhibit medium to large positive correlations. We demonstrate a use-case for latent difficulty example parameters, namely training set filtering, and show that using difficulty to sample training data outperforms baseline methods. Finally, we highlight cases where human expectation about example difficulty does not match difficulty as estimated from the machine RPs.

5.1.1 Motivation

What is the most difficult example in the Stanford Natural Language Inference (SNLI) data set [Bowman et al., 2015] or in the Stanford Sentiment Treebank (SSTB) [Socher et al., 2013]? *A priori* the answer is not clear. How does one quantify the difficulty of an example and does it pertain to a specific model, or more generally?

There has been much recent work trying to assess the quality of data sets used for NLP tasks [Sakaguchi and Van Durme, 2018, Kaushik and Lipton, 2018]. In particular, a common finding is that different examples within the same class have very different qualities such as difficulty, and these differences affect models' performance. For example, one study found that a subset of reading comprehension questions were so difficult as to be unanswerable [Kaushik and Lipton, 2018]. In addition, we have shown that the difficulty of specific examples was found to be a significant predictor of whether a model would classify the example correctly (Chapter 3).

While a number of methods exist for estimating difficulty, in this work we focus on Item Response Theory (IRT) [Baker, 2001, Baker and Kim, 2004], a widely used method in psychometrics. IRT models fit parameters of examples such as difficulty based on a large number of annotations ("response patterns" or RPs), typically gathered from a human population ("subjects"). It has been shown to be an effective way to evaluate and analyze NLP models with respect to human populations (Chapters 2 and 3).

While IRT models are designed to be learned with human RPs for at most 100 examples, data sets used in machine learning, particularly for training deep neural networks (DNNs), are on the order of tens or hundreds of thousands of examples or more. It is not possible to ask humans to label every example in a data set of that size. In this chapter we hypothesize that IRT models can be fit using RPs from artificial crowds of DNNs as inputs, thereby removing the expense of gathering human RPs. Recent work has shown that DNNs encode linguistic knowledge [Tenney et al., 2019b, Tenney et al., 2019a] and can reach or surpass human-level performance on classification tasks [Lake et al., 2015]. In addition, generating IRT data with deep learning models is much cheaper compared to employing human annotators.

We demonstrate that learned parameters from IRT models fit with artificial crowd data are positively correlated with parameters learned with human data for small data sets. We then use variational inference (VI) methods [Jordan et al., 1999, Hoffman

et al., 2013] to fit a large-scale IRT model. Using VI allows us to scale IRT models to deep-learning-sized data sets. Finally, we show why learning such models is useful by demonstrating how learned difficulties can improve training set subsampling.

Our contributions are as follows: (1) We show that IRT models can be fit using machine RPs by comparing example parameters learned from human and from machine RPs for two NLP tasks; (2) we show that RPs from more complex models lead to higher correlations between parameters from human and machine RPs; (3) we demonstrate a use-case for latent difficulty example parameters, namely training set filtering, and show that using difficulty to sample training data outperforms baseline methods; (4) we provide a qualitative analysis of examples with the largest human-machine disagreement in terms of difficulty to highlight cases where human intuition is inconsistent with model behavior.

These results provide a direct comparison between humans and machine learning models in terms of identifying easy and difficult examples. They also provide a foundation for large-scale IRT models to be fit by using ensembles of machine learning models to obtain RPs instead of humans, greatly reducing the cost of data-collection.[1]

5.2 Data and Models

Here we describe the data sets used to conduct our experiments, as well as the DNN model architectures for both generating response patterns and conducting our training set filtering experiment.

For NLP we again experiment with the SNLI and SSTB data sets (§2.2.1). To test the applicability of our methods to other domains, we also experiment with two data sets from the computer vision community:

[1]Code for IRT model fitting is available at `https://github.com/jplalor/py-irt`.

5.2.1 MNIST

The MNIST data set [LeCun et al., 1998] is a data set of handwritten digits from 0 to 9. It includes 60,000 training examples and 10,000 test examples, and is regularly used to benchmark new machine learning models. We split the training set and use the first 50,000 examples for training and the last 10,000 examples as a validation set. With MNIST, we use a straightforward convolutional neural network (CNN) architecture [LeCun et al., 1995] with two convolutional layers and two fully connected layers with ReLU activations [Nair and Hinton, 2010]. Max-pooling layers [Krizhevsky et al., 2012] are included after both convolutional layers, and there is a dropout layer between the first and second fully connected layers [Srivastava et al., 2014]. Models were trained for 100 epochs using stochastic gradient descent (SGD) with a learning rate of 0.01 and momentum of 0.5.

5.2.2 CIFAR

The CIFAR data set [Krizhevsky and Hinton, 2009] is another popular image recognition data set where each image is associated with 1 of 10 classes. It is a labeled subset of the 80 million tiny images data set [Torralba et al., 2008]. Class labels include "dog," "automobile," and "truck." CIFAR consists of 50,000 training examples and 10,000 test examples. We split the training set and use the first 40,000 for training and the last 10,000 as a validation set. For the CIFAR experiments we use the VGG-16 CNN model [Simonyan and Zisserman, 2015], a deep CNN model that has shown impressive performance on image recognition tasks, including CIFAR. CIFAR models were trained for 1000 epochs using SGD with a learning rate of 0.01, momentum of 0.9, and weight decay of 0.0005. MNIST and CIFAR models were implemented in PyTorch [Paszke et al., 2017].

5.2.3 Human RP Data

The human RP data sets for SNLI and SSTB were previously collected from Amazon Mechanical Turk (AMT) workers (Chapter 2). For a randomly selected sample of examples from SNLI and SSTB, new labels were gathered from 1000 AMT workers (Turkers). Each Turker labeled each example, so that for each example there were 1000 new labels. For each Turker, a RP was generated by grading the provided labels against the known gold-standard label.

5.2.4 Building an Artificial Crowd

As mentioned earlier, it is not feasible to have humans provide RPs for data sets used to train DNN models. Can we instead use RPs from DNNs? We trained an ensemble of DNN models with varying amount of training data to simulate an artificial crowd so that enough responses were obtained to fit the IRT models. The goal here is not to build an ensemble of DNNs to surpass current classification state of the art results, but instead to test our hypothesis to determine if machine RPs can fit IRT models that can benefit NLP tasks.

For this work we tested with two deep learning models: LSTM and NSE (§3.2.2). Specifically, we trained 1000 LSTM models for NLI classification using the SNLI data set and 1000 LSTM models for binary SA classification using the SSTB data set [Bowman et al., 2015, Socher et al., 2013]. The SNLI model consists of two LSTM sequence-embedding models [Hochreiter and Schmidhuber, 1997], one to encode the premise and another to encode the hypothesis. The two sentence encodings are then concatenated and passed through three tanh layers. Finally, the output is passed to a softmax classifier layer to output class probabilities. For SSTB, we used a single LSTM model without the concatenation step. The models were implemented in DyNet [Neubig et al., 2017]. Models were trained with SGD for 100 epochs with a learning rate of 0.1, and validation set accuracy was used for early stopping.

For each model m_i, we randomly sampled a subset of the task training set, x^i_{train}. We corrupted a random selection of training labels by replacing the gold standard label with an incorrect label. For each model-training set pair, we trained the model, used the held out validation set for early stopping, and wrote the model's graded (correct/incorrect) outputs to disk as that model's RP. The set of RPs for all models is our input data for the IRT models.

We also looked at a more complex model to determine if the learned parameters would differ given the different model architectures. For our more complex model we used the Neural Semantic Encoder model (NSE), a memory-augmented recurrent neural network [Munkhdalai and Yu, 2017]:

$$
\begin{aligned}
o_t &= f_{\text{r}}^{LSTM}(x^t) \\
z_t &= softmax(o_t^\top M_{t-1}) \\
m_{r,t} &= z_t^\top M_{t-1} \\
c_t &= f_{\text{c}}^{MLP}(o_t, m_{r,t}) \\
h_t &= f_{\text{w}}^{LSTM}(c_t) \\
M_t = M_{t-1}(\mathbf{1} &- (z_t \otimes e_k)^\top) \\
&+ (h_t \otimes e_t)(z_t \otimes e_k)^\top
\end{aligned}
$$

where f_r^{LSTM} is the read function, f_c^{MLP} is the composition function, f_w^{LSTM} is the write function, M_t is the external memory at time t, and $e_l \in R^l$ and $e_k \in R^k$ are vectors of ones.

The goal with the data set restriction and label corruption was to build an ensemble of models with widely varying performance on the SNLI test set. Training with different training set sizes and levels of noise corruption means that certain models will perform very well on the test set (large training sets and low label corruption) while others will perform poorly (small training sets and high label corruption). This way we

will get a variety of response patterns to simulate performance on the task across a spectrum of ability levels. While we could have modified the networks in any number of ways (e.g. changing layer sizes, learning rates, etc.), modifying the training data is a straightforward method for generating a variety of response patterns, and has been shown earlier to have an impact on performance in terms of example difficulty (Chapter 3). Further investigations of network modifications is left for future work.

5.3 Methods

We conduct the following experiments: (i) a comparison of IRT parameters learned from human and machine RP data, using existing IRT data sets as the baseline for comparison, (ii) a comparison between MML and VI parameter estimates, and (iii) a demonstration of the effectiveness of learned IRT parameters via training data set selection experiments.

5.3.1 Validating Variational Inference

Before using VI to fit IRT models for DNN data, we must first show that VI produces estimates similar to traditional methods. This was established in prior work on synthetic data [Natesan et al., 2016]. Here we compare them on an existing human data set.

A traditional Rasch model was fit with both MML and VI. MML was implemented in the R package mirt [Chalmers et al., 2012] and VI in Pyro [Bingham et al., 2019], a probabilistic programming language built on PyTorch [Paszke et al., 2017] that implements typical VI model fitting and variance reduction [Kingma and Welling, 2014, Ranganath et al., 2014]. We calculate the root mean squared difference (RMSD) between MML and VI estimates for subject and example parameters. Our expectation is that the RMSD will be sufficiently small to confirm that the VI parameters are

similar enough to those learned by MML, since we will not be able to use MML when we attempt to scale up to larger data sets.

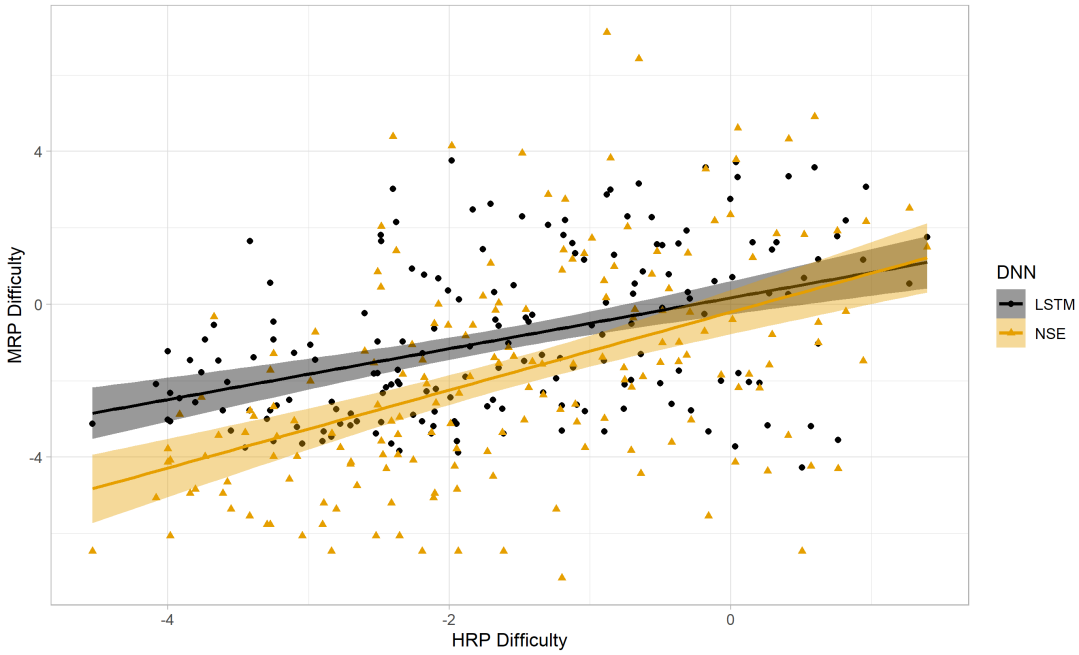
5.3.2 Human Machine Correlation

We further compare example difficulty parameters learned from machine RPs to those learned from human RPs. These two sets of parameters cannot be compared directly as they can only be interpreted in reference to their respective subject populations. Instead, we compute the correlation between these two sets of parameters to see whether examples that are easy for humans are also easy for machines. We fit two Rasch models, one with existing human RPs, and one with the machine RPs. Both models were fit with MML using the mirt R package [Chalmers et al., 2012]. Learned example difficulty parameters were extracted and compared via Spearman ρ rank order correlations.

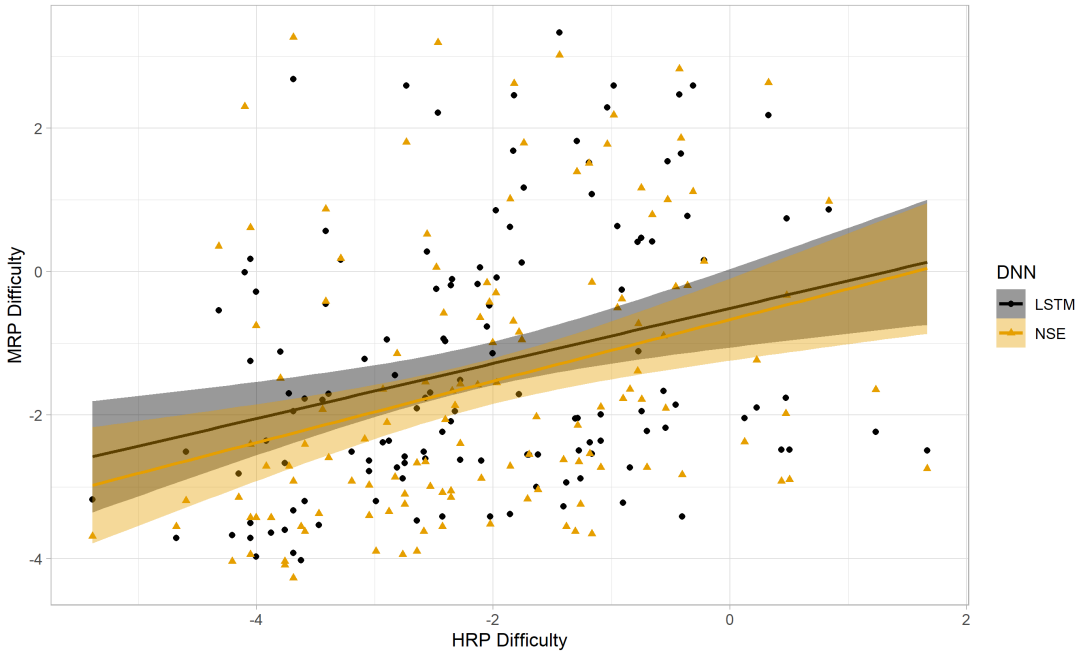
5.3.3 Training Set Subsampling

To demonstrate the usefulness of the learned IRT parameters, we next describe a downstream use case: training set filtering for more efficient learning. Can we maintain model performance by removing the easiest and/or hardest examples from the training set? Once difficulty parameters for each data set were learned, we trained a new DNN model using only a subset of the original training data. We trained a number of models, each with a different cutoff in terms of training data to observe how generalization was impacted in each case.

We looked at 4 filtering strategies (in each case d is the example difficulty threshold): (i) absolute value inner (AVI), where all training examples with $|b_i| < d$ were retained, (ii) absolute value outer (AVO), where all training examples with $|b_i| > d$ were retained, (iii) an upper bound (UB), where examples with $b_i < d$ were retained, and (iv) a lower bound (LB), where examples with $b_i > d$ were retained. These methods were compared against two baselines that consider the percentage of models that label an



(a) NLI



(b) SA

Figure 5.1: Comparison of learned example difficulty parameters for human (x-axis) and machine data (y-axis) for NLI (Fig. 5.1a) and SA (Fig. 5.1b). Spearman ρ (NLI): 0.409 (LSTM) and 0.496 (NSE). Spearman ρ (SA): 0.332 (LSTM) and 0.392 (NSE).

example correctly ($0 \leq pc \leq 1$) as an inexpensive proxy for difficulty: (i) percent-correct upper bound (PCUB), where examples with $pc_i < d$ were retained, and (ii) percent-correct lower bound (PCLB), where examples with $pc_i > d$ were retained. Setting an upper bound on difficulty (UB) is similar to setting a lower bound on percent correct (PCLB) (i.e., we are excluding the hardest examples from training). Similarly, setting a lower bound on difficulty (LB) is analogous to setting an upper bound on percent correct (PCUB) in that they both exclude the easiest examples from training.

Each of the filtering strategies have arguments in favor of their potential effectiveness. AVI includes "average" examples in terms of training examples, none that are too easy or too difficult. AVO is the opposite, where only the easiest and most difficult examples are retained, so that the extremes for each class can be learned. UB ensures that those examples that are too difficult are not included, and LB ensures that the examples that are too easy are not included so that the model doesn't spend time learning very easy examples.

5.4 Results

5.4.1 Human Machine Model Correlations

We first look at the results of our human-machine model comparison (Figures 5.1a and 5.1b). As an upper bound for correlations, we split the human annotation data in half for both SNLI and SSTB, fit two IRT Rasch models, and calculated the correlation between the learned parameters. Spearman ρ values were 0.992 and 0.987 for SNLI and SSTB examples, respectively.

For both SNLI and SSTB, we find a positive correlation between the example difficulties of IRT models fit using human and machine RPs. In addition, the more complex NSE model has consistently a higher correlation with the human-learned difficulty parameters than the LSTM model. This suggests that creating more complex

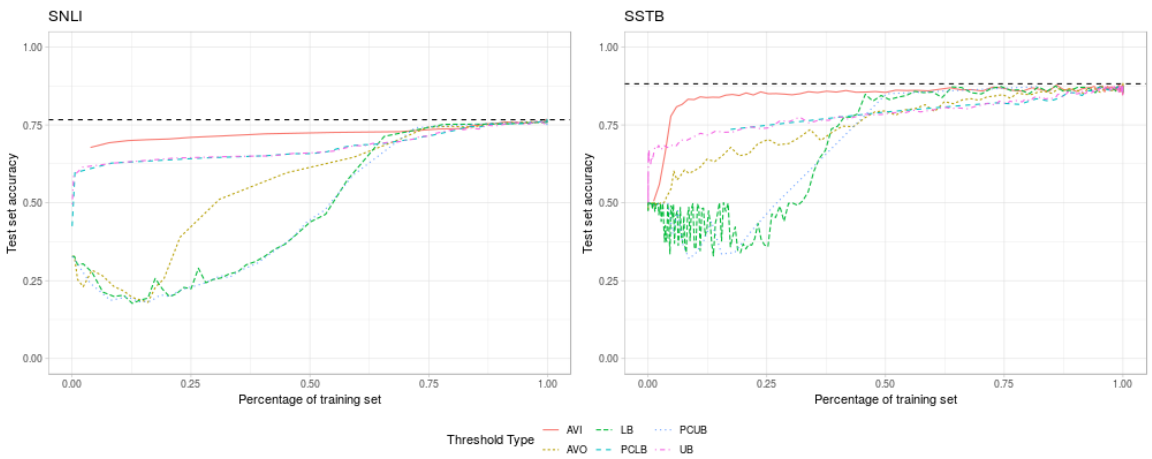


Figure 5.2: Test set accuracy by filtering strategy for NLI (left) and SA (right) plotted against percentage of training data retained. In both tasks filtering using the AVI strategy is most efficient in terms of high accuracy for small training set sizes.

DNN architectures has bearing on how the model identifies difficult examples with regards to human expectations.

The correlation is not perfect, and we would argue that this is an expected and encouraging result. A close to perfect correlation would indicate that the DNN models and the human population agree closely on the difficulty ranking for the data sets and would be an incredible finding and evidence for the argument that DNN models encode human knowledge well, at least with respect to the difficulty of specific examples. This of course is not true, and the positive but not perfect correlation coefficients indicate this as such. That said, it is encouraging that the positive correlation exists. One would expect that training ensembles of more sophisticated NLP models such as BERT [Devlin et al., 2019] would further increase correlation scores.

5.4.2 Learning IRT Models with VI

Our next goal was to determine if VI could be used to fit IRT models and confirm prior work to that effect [Natesan et al., 2016]. The RMSDs between MML and VI estimates were 0.158 and 0.154, respectively, for the difficulty and ability parameters. Learned parameters are very similar between the two methods, which is to be expected.

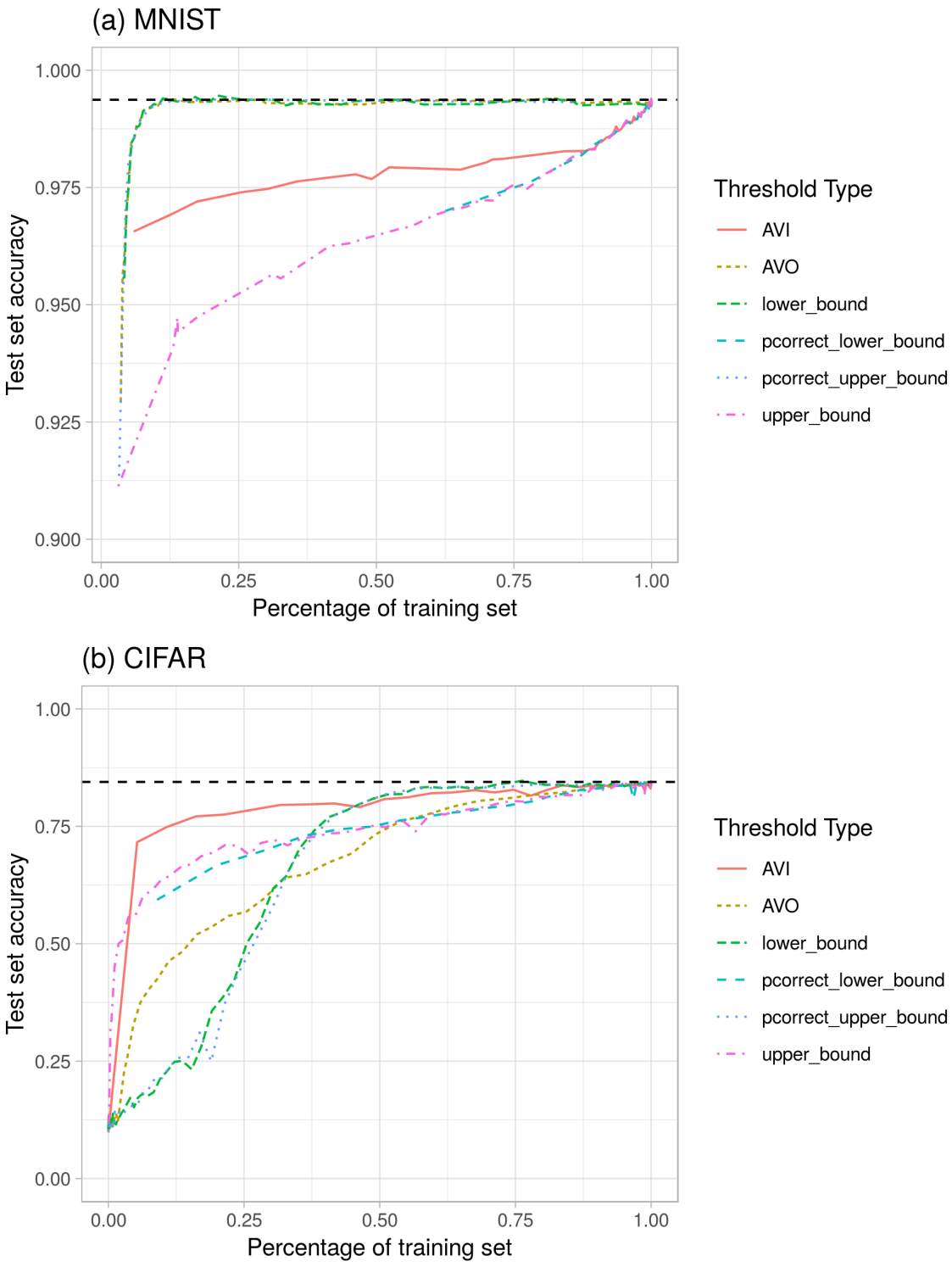


Figure 5.3: Test set accuracy for MNIST and CIFAR for each filtering strategy plotted as a function of the percentage of training data retained.

This echos the results of prior work showing that VI is a good alternative to traditional MML methods for learning IRT models [Natesan et al., 2016]. This result holds not only with synthetic data, as was used in the prior work, but also with human data collected for the development of an actual IRT test.

5.4.3 Data Filtering

Finally we consider training new DNN models on the filtered training data sets, restricted according to latent difficulty and the strategies described above (Figure 5.2). The horizontal dotted lines in each plot represent the test set accuracy for a model trained with the full training data set. For both SNLI and SSTB, the AVI strategy of selecting "average" examples leads to very good test set accuracy scores with less than 25% of the original training data. This shows that the strategy of selecting training data in terms of average difficulty, and gradually adding easier and harder examples at the same time provides examples that allows trained models to generalize well. For both tasks, there is a large number of examples that are very easy in terms of latent difficulty (Figure 5.4). Sampling with AVI avoids selecting too many examples that are too easy and instead selects examples that are of average difficulty for the task, which may be better for learning. In both cases LB and PCUB are the least effective strategies, indicating that it is not enough to only include the most difficult examples.

The plots show that PCUB and LB provide very similar results, as do PCLB and UB, which is to be expected. Difficulty parameters learned from IRT are very similar to metrics such as percent correct, but as the plots show are not exactly the same. Differences in RPs (i.e. which specific examples were answered correctly/incorrectly) have an effect on example difficulty that is not captured by calculating percent correct.

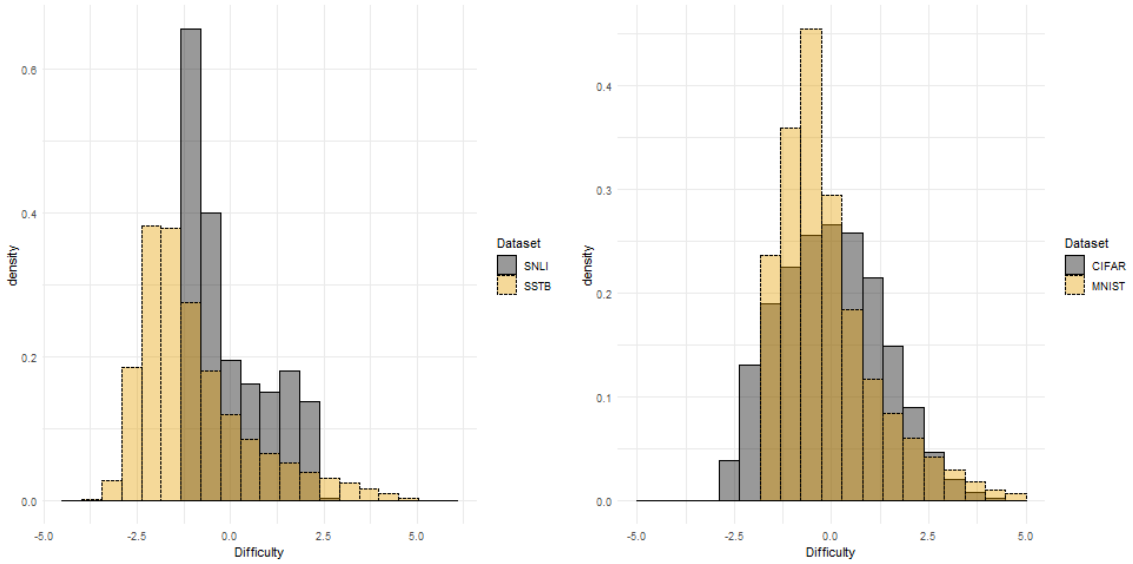


Figure 5.4: Density plot of learned difficulties for SNLI and SSTB (left) and MNIST and CIFAR (right) data sets.

It is worth noting here that the filtering strategy we used did not take class labels into consideration.[2] The only determining factor as to whether a training example was included was the learned difficulty parameter b_i, which led to class imbalances in the training set. This imbalance, however did not seem to have a significant negative effect in terms of performance.

Figure 5.3 shows results of the training data filtering experiments for MNIST and CIFAR, respectively. Note that for MNIST, test set accuracy was above 90% even for very small percentages of the training set, and therefore the MNIST plot y axis is truncated to show variations more clearly (Fig. 5.3). For both data sets, removing up to 50% of the training data according to learned difficulty maintains test set accuracy within a few percentage points of the baseline. For MNIST, baseline accuracy is maintained with as little as 15% of the training data.

[2]This is true for only the filtering step. Class labels are needed for learning the difficulty parameters needed for filtering (§1.3).

Strategy	% of Training Data		
	0.1%	1%	10%
Random (reported)	82.1	85.2	**88.4**
Random (small batch)	81.79	84.90	88.32
Lower-bound	43.68	41.56	39.89
Upper-bound	81.62	80.46	79.06
AVI	**82.44**	**85.44**	86.73
AVO	43.60	42.05	40.81

Table 5.1: Dev accuracy results for MT-DNN model with different training set sampling strategies.

For MNIST, AVO and LB are more effective filtering strategies than AVI and UB, while AVI is the most effective for CIFAR. For MNIST, relative variance within the class is small. That is, even the hardest "3" still looks like a single numerical digit. Therefore it is unnecessary to include a large number of examples of average difficulty in order to learn a particular class, making AVO an effective strategy. Similarly, the easiest examples in a class can be ignored in favor of more challenging ones (LB).

For CIFAR, on the other hand, there is much more variance within each class. In these cases the easiest and hardest examples may truly be outliers in terms of the class. Therefore the DNN models would require more examples from the middle of the difficulty distribution to learn a representation of the class (AVI). That said, LB is the least effective strategy in both cases, indicating that it is not enough to only include the most difficult examples.

As with the SNLI experiments, the filtering strategy we used did not take class labels into consideration. More advanced sampling strategies that maintain training set distribution or sample data using a Bayesian approach are left for future work.

As an additional experiment, we used the learned difficulty parameters to compare data sampling strategies for a state-of-the-art NLI model, MT-DNN [Liu et al., 2019]. We sampled training data for SNLI at several intervals (0.1%, 1%, 10%) and trained the MT-DNN model with the sampled data. We trained each model, as well as the

Label	Premise	Hypothesis	Difficulty
Ent.	Two men and a woman are inspecting the front tire of a bicycle.	There are a group of people near a bike.	-3.675
	A street vendor selling cupcakes.	There is a person outside in this picture	-3.506
	A young boy in a red shirt plays on a mini-trampoline in a grassy field	Someone is outside.	-3.483
	This is nice place to relax and chat.	the place is nice	2.235
	Neck and neck to the finish line, every competitor has been training for this race.	The competitors have trained very hard and are all very close to the finish line.	2.759
	A girl in a newspaper hat with a bow is unwrapping an item.	The girl is going to find out what is under the wrapping paper.	3.144
Cont.	Two dogs playing in snow.	a cat sleeps on floor	-4.014
	Girls playing soccer competitively in the grass.	Nobody is playing soccer.	-3.558
	The backside of a woman leaning against the guard rail of a passenger boat looking out at the open ocean.	The woman is driving her car on the highway.	-3.407
	A rider mid-jump on a snowmobile during a race.	A snowboarder in mid-air during a race.	3.639
	A man and woman walking away from a crowded street fair.	There are a group of men walking together.	3.658
	Man sweeping trash outside a large statue.	A man is on vacation.	3.766
Neut.	People sitting in chairs with a row flags hanging over them.	a family reunion for fourth of July	-3.603
	Two men together, one watching, one resting.	Two men are together because they are friends.	-3.446
	two girls on a bridge dancing with the city skyline in the background	The girls are sisters.	-3.385
	A wielder works on wielding a beam into place while other workers set beams.	The wielder is working on a building.	2.864
	Two soccer players on the field running into each other.	There are two people colliding and falling.	3.422
	A group of dancers are performing.	The audience is silent.	3.798

Table 5.2: The easiest and hardest examples judged by machine responses for each class in the SNLI test data set.

random sample baseline, using the publicly available MT-DNN code.[3] Results are

[3] `https://github.com/namisan/mt-dnn`

Task	Label	Example Text	Difficulty ranking		
			Humans	LSTM	NSE
SNLI	Contradiction	*P:* Two dogs playing in snow. *H:* A cat sleeps on floor	168	1	5
	Entailment	*P:* A girl in a newspaper hat with a bow is unwrapping an item. *H:* The girl is going to find out what is under the wrapping paper.	55	172	176
SSTB	Positive	Only two words will tell you what you know when deciding to see it: Anthony. Hopkins.	9	103	110
	Negative	...are of course stultifyingly contrived and too stylized by half. Still, it gets the job done–a sleepy afternoon rental.	128	46	41

Table 5.3: Examples from the SNLI and SSTB data sets where the ranking in terms of difficulty varies widely between human and DNN models. In all cases difficulty is ranked from easy to hard (1=easiest).

Figure 5.5: The easiest (first and third rows) and hardest (second and fourth rows) examples in the MNIST and CIFAR test sets.

reported in Table 5.1. Note that we report two random baselines: (i) those reported in the original work, which were obtained by training the MT-DNN model with a batch size of 32. Due to GPU resource constraints we had to train each MT-DNN model with a batch size of 8, and therefore report our reproduced random baseline results that we obtained as well ("Random (small batch)"). For very small samples of data, the AVI strategy outperforms random sampling and all other methods as well. As more data is sampled, the random models perform better. This indicates that a more advanced sampling strategy that starts with AVI then incorporates outliers (very easy/hard examples) at certain thresholds may improve learning as well.

5.5 Analysis

5.5.1 Qualitative Evaluation of Difficulty

Table 5.2 shows examples of premise-hypothesis sentence pairs from SNLI with the learned difficulty parameter from the machine RP IRT model. The easy sentence pairs for each class seem to be very obvious, whereas the most difficult examples are difficult due to ambiguity. For example, the hardest contradiction example could be classified as neutral instead of contradiction. It could be the case that the man is sweeping while on vacation, though it isn't likely. The hypothesis doesn't directly contradict the premise like the easy example does (cats instead of dogs, sleeping instead of playing).

We were also able to show that the learned difficulty parameters are interpretable for image tasks such as MNIST and CIFAR. Figure 5.5 shows the easiest and hardest examples in the test data sets. For a certain class, there are examples that one may consider more difficult than others, due to noise or irregular lines (in the case of MNIST), and this is reflected in the learned difficulty parameters. As we can see, there is interpretability in the learned difficulty parameter b_i. The difference between the easiest and hardest examples in the MNIST test set for each digit is clear. The easiest examples are very much prototypical examples of their specific digit, while the

hardest examples for each digit are outliers and in some cases (e.g. 3 and 8) are hard to distinguish from certain other digits. For CIFAR the differences are present but more subtle because the variation in the images is greater. For the hardest examples it seems that the difficulty arises mostly from the subject of the image being non-typical for the class, either according to color or orientation. For example, the hardest "car" is a car in a rotary lift, which is not common for cars, and the hardest "ship" is sitting on land instead of water. The hardest "frog" is blue, and the hardest "dog" is wearing an orange sweater. These are not consistent with the typical cases for each class, which may be the reason that the DNN models do not perform well with regards to labeling them.

5.5.2 Analysis of Differences

An interesting question comes up as a result of the less-than-perfect correlation scores (§5.4.1): Where are the differences? To examine these more closely we identified those examples from the data sets where the rank order was most different between the human- and machine-response pattern models (Table 5.3). That is, we calculated the absolute difference in ranking between the human model and the DNN model, and selected those where that value was highest. The average absolute difference in ranking was around 40 for the SNLI task and around 30 for SSTB, for both the LSTM and NSE ensembles.

We can see interesting patterns in the discrepancies. For SNLI, the easiest sentence pair for the LSTM model (which is also very easy for the NSE model) is one of the hardest for humans (Table 5.3, row 1). Upon inspection of the gathered labels, the high difficulty comes from the fact that there were many Turkers who labeled the data as neutral and also many who labeled it as contradiction.

On the other hand, an example that is easy for humans but difficult for the DNN models (Table 5.2, row 2) requires more abstract thinking than the earlier example.

The humans are able to infer that because the girl is unwrapping an example, she will discover what is under the wrapping paper when the unwrapping is complete. The models find this pair to be one of the most difficult in the data set.

For SSTB, we see similar patterns (Table 5.3, rows 3-4). For humans, one of the easiest review snippets is clearly positive (row 3), mainly because we know who Anthony Hopkins is and know how to rate his quality as an actor. However for the DNN models, the text itself does not have a lot of positive or negative signal and therefore the example is considered very difficult. On the other hand, the last example is very difficult for humans (row 4), possibly due to the relatively neutral text. However, for the DNN models certain terms such as "stultifyingly contrived" may signal a more negative review and lead to the example being easier.

In both cases, it is not clear if there is a "gold standard" for difficulty. Estimating difficulty using IRT relies on responses from a group of humans or an ensemble of models, and the resulting difficulty estimates may be biased based on who or what provides the labels. Human intuitions or model architecture decisions impact the response patterns collected, which in turn affect the learned parameters. An investigation into what upstream information drives downstream effects such as learned difficulty is an interesting and important direction for future work.

5.6 Conclusion

We have described how large-scale IRT models can be trained with DNN response patterns using VI. Learning the difficulty parameters of examples and the ability parameters of DNN models allows for more nuanced interpretation of model performance and enables us to filter training data so that DNN models can be trained on less data while maintaining generalization as measured by test set performance. IRT models with machine RPs can be fit not only for NLP data sets but also data sets in other machine learning domains such as computer vision.

One limitation of this work is the up-front cost of generating RPs from the DNN ensemble. However, the cost of running a large number of DNN models to generate response pattern data is significantly less than the cost of obtaining those labels from human annotators in two ways. First, the monetary cost of asking thousands of humans to label tens or hundreds of thousands of images or sentence pairs is prohibitive. Second, since the response patterns require that a single individual provide labels for all (or most) of the data set, each individual would need to label a huge number of examples. Each individual would most likely get bored or burned out and the quality of the labels would suffer.

That said, consider for example a large company (or research lab) that runs hundreds or thousands of experiments each day on some internal data set. Many of the experiments would not lead to significant improvements in model performance, and the outputs from those experiments would be discarded. With the methods proposed here those outputs can be used to learn the latent parameters of the data to focus in on what exactly is working well and what isn't with respect to the models being tested and the data used to train them. Using the previously discarded data to learn IRT models and estimate latent difficulty and ability parameters can be used to improve a variety of tasks such as model selection, data selection, and curriculum learning strategies.

IRT models assume difficulty is a latent parameter of the examples and can be estimated from response pattern data. Difficulty is directly linked to subject ability, in contrast to heuristics such as sentence length or word rarity. Certain examples may be easy or difficult for a variety of reasons. With the methods presented here, an interesting direction for future work is to further examine why certain examples are more difficult than others.

We have shown that it is possible to fit IRT models using RPs from DNN models. This work also opens the possibility of fitting IRT models on much larger data sets.

By removing the human bottleneck, we can use ensembles of DNN models to generate RPs for large data sets (e.g. all of SNLI or SSTB instead of a sample). Having difficulty and ability estimates for machine learning data sets and models can lead to very interesting work around such areas as active learning, curriculum learning, and meta learning.

CHAPTER 6

DYNAMIC DATA SELECTION FOR CURRICULUM LEARNING VIA ABILITY ESTIMATION

6.1 Introduction

Curriculum learning is a popular and well-studied method for machine learning model training. However, most methods rely on heuristics to estimate the easiness or difficulty of training examples when building curricula. In this chapter we show that modeling difficulty using psychometric methods is more effective for curriculum learning than heuristic measures of difficulty such as sentence length. We introduce Dynamic Data selection for Curriculum Learning via Ability Estimation (DDaCLAE), a curriculum learning strategy that probes model ability at each training epoch to select the best training examples at that point in time. DDaCLAE adds data at a rate commensurate with the model's capability, in contrast to scheduled curricula that add data at a predetermined rate. Experimental results demonstrate that DDaCLAE is more efficient and effective than existing curriculum learning methods, improving test set accuracy while reducing training set size by up to 88%.

6.1.1 Motivation

Curriculum learning, the process of training a model by showing easy examples first and gradually adding more difficult examples, can speed up learning and improve generalization in machine learning models, a result that has also been shown in humans [Bengio et al., 2009, Amiri et al., 2017, Platanios et al., 2019]. The basic premise is that machine learning models are trained according to a curriculum that sorts training examples according to difficulty. At first, the model is trained with only

the easiest examples, and more difficult examples are gradually added according to some schedule. A benefit of curriculum learning methods is that model convergence can often be faster than a baseline model trained without a curriculum [Bengio et al., 2009, Platanios et al., 2019]. With the size of machine learning models and data sets continuing to grow, and with better understanding of the impact of model training on the environment, there is a growing need for efficient model training [Strubell et al., 2019].

A major drawback of existing curriculum learning techniques is that they rely on heuristics to measure the difficulty of data, and either ignore the competency of the model at its present state or rely on heuristics there as well. For example, often for natural language processing (NLP) tasks, sentence length is considered a proxy for difficulty [Bengio et al., 2009, Platanios et al., 2019]. Similarly, the number of objects in an image has been used as a proxy for difficulty in an image recognition task [Bengio et al., 2009]. These heuristics can be useful but have limitations. First, a model's notion of difficulty may not align with the heuristic imposed by a human developing the model. It could be the case that examples that appear difficult for the human are in fact very easy for the model to learn. Second, the heuristic chosen may not actually be a proxy for difficulty. For example, often times sentence length is used as a proxy for difficulty in NLP tasks. However, depending on the task, long sequences could signal easier or harder examples, or have no signal at all with regard to difficulty.

Competency was recently introduced as a mechanism to determine when new examples should be added to the training data [Platanios et al., 2019]. However, in that work competency is assumed to be a monotonically increasing function of a pre-determined initial competency c_0. Competency is not evaluated during training. Ideally, model competency would be measured at each training epoch, so that the

training data could be appropriately matched with the model at a given point in the training.

As we have shown in earlier chapters, it is possible to estimate both the difficulty of examples and the ability of deep learning models as latent variables based on model performance using Item Response Theory (IRT). IRT is a well-studied methodology in the psychometric literature for test set construction and subject evaluation [Baker and Kim, 2004]. A typical IRT model will estimate latent parameters such as difficulty for examples under consideration for inclusion in a test set and a latent ability parameter for individuals taking the test. This is done by administering a test to a large number of human subjects, collecting and grading their responses as correct or incorrect, and using the student-response data matrix to estimate the latent traits of the data. Once learned, these parameters can be used to estimate latent ability parameters of future test-takers, based on their graded responses to the examples. IRT has not seen wide adoption in the machine learning community, primarily due to the fact that fitting IRT models requires a large amount of human annotated data for each example. Because one can learn example difficulty and subject ability together, IRT is an interesting framework to consider for the problem of curriculum learning.

We propose Dynamic Data selection for Curriculum Learning via Ability Estimation (DDaCLAE, pronounced "day-clay"), a novel curriculum learning framework that uses the estimated ability of a model at a specific point in the training process to identify appropriate training data. At each training epoch, the latent ability of the model is estimated. Based on this estimate, only training data that the model has a reasonable chance of labeling correctly is included in training. As the model improves, the estimated ability will improve, and more training examples will be added.

Our contributions are as follows: (i) we propose a novel curriculum learning framework, DDaCLAE, which automatically selects training data based on the estimated ability of the model, (b) we show that model training using DDaCLAE leads to faster

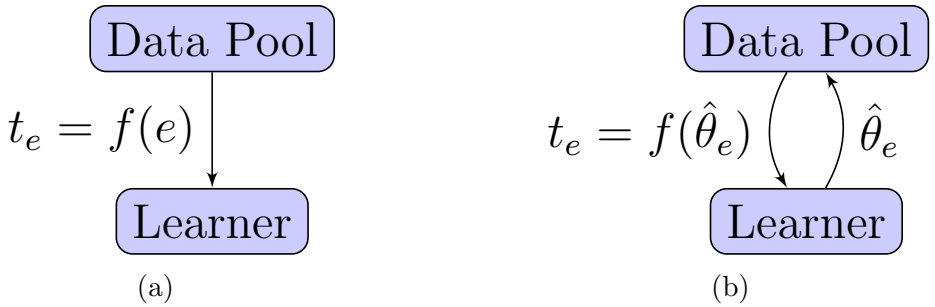


Figure 6.1: (6.1a) A typical curriculum learning framework, where examples are added at each epoch according to a static monotonically-increasing learning schedule. (6.1b) DDaCLAE estimates ability at each training epoch to dynamically select appropriate training data given the model's current ability.

convergence and better performance than traditional training and baseline curriculum learning methods, (c) we analyze DDaCLAE to show why certain training examples hurt instead of help generalization. This is the first work to learn a model competency during training that is directly comparable to the difficulty of the training data pool.

6.2 Methods

We first describe a typical curriculum learning framework. We then introduce IRT for those unfamiliar with the methodology, specifically the one-parameter logistic (1PL) model, also referred to as the Rasch model [Rasch, 1960, Baker and Kim, 2004]. Learning IRT models for machine-learning scale data sets with variational inference methods is then described, and finally we introduce the Dynamic Data selection for Curriculum Learning via Ability Estimation model (DDaCLAE) for probing model ability to select training examples.

6.2.1 Curriculum Learning

In a traditional curriculum learning framework, training data examples are ordered according to some notion of difficulty, and the training set shown to the learner is augmented at a set pace with more and more difficult examples (Fig. 6.1).

Algorithm 3 DDaCLAE

Input: (X, Y), model ϕ, D
Output: Learned model ϕ

while True **do**
    $\hat{Y} = \phi(X)$
    $\hat{\theta}_e = score(Y, \hat{Y}, D)$
    $X_e, Y_e = \{(x, y) : d_x < \hat{\theta}_e\}$
    $train(\phi, X_e, Y_e)$
end while
procedure SCORE($Y, \hat{Y}, D$)
    $Z = \forall_{y \in Y} \mathbf{I}[y_i = \hat{y}_i]$
    $\hat{\theta}_e = \arg\max_\theta p(Z|\theta, b)$
    return $\hat{\theta}_e$
end procedure

Typically, the model's current performance is not taken into account. Recent work has incorporated a notion of competency to curriculum learning [Platanios et al., 2019]. In that work the authors structure the rate at which training examples are added based on an assumption that model competency is modeled by either a linear or root function of the training epoch. However there are two issues with such an approach. First, this notion of competency is artificially rigid. If a model's competency improves quickly, more data cannot be added more quickly because the rate is predetermined. On the other hand, if a model is slow to improve, it may struggle because more data is being added too quickly. Second, the formulation of competency proposed by the authors reduces to a competency-free curriculum learning strategy with a tunable parameter for inclusion speed. Once this parameter is set, there is no check of model ability during training to assess competency and update training data examples. We do away with the heuristics and instead measure difficulty and competency directly.

6.2.2 Dynamic Data selection for Curriculum Learning via Ability Estimation

We propose DDaCLAE, where training examples are selected dynamically at each training epoch based on the estimated ability of the model at that epoch. With

DDaCLAE, model ability can be estimated according to a well-studied psychometric framework as opposed to heuristics. The estimated ability of the model at a given epoch e ($\hat{\theta}_e$) is on the same scale as the difficulty parameters of the data, so there is a principled approach for selecting data at any given training epoch.

The first step of DDaCLAE is to estimate the ability of the model using the scoring function (§1.3.7). To do this we use the full training set, but crucially, only to get response data, not to update parameters (i.e., no backward pass). We do not use a held out development set for estimating ability because we do not want the development set to influence training. In our experiments the development set is only used for early stopping. Model outputs are obtained for the training set, and graded as correct or incorrect as compared to the gold standard label. This response pattern is then used to estimate model ability at the current epoch ($\hat{\theta}_e$).

Once ability is estimated, data selection is done by comparing estimated ability to the examples' difficulty parameters. Each example in the training pool has an estimated difficulty parameter (b_x). If the difficulty of an example is less than or equal to the estimated ability, then the example is included in training for this epoch. Examples where the difficulty is greater than estimated ability are not included.

With DDaCLAE, the training data size does not have to be monotonically increasing. If a model's performance suffers as a result of adding data too quickly, then this will be reflected in lower ability estimates, which leads to less data selected in the next epoch. This avoids a scenario where data is added too quickly at the expense of learning the easier examples. At the same time, if estimated model ability is high, then more data can be added more quickly, without artificially slowing down learning.

Algorithm 3 shows all of the steps for DDaCLAE. Code implementing DDaCLAE is included as supplemental material and will be released upon publication.

6.3 Data and experiments

We experiment with four data sets, two from vision (§5.2) and two from NLP (§2.2.1), to demonstrate the effectiveness of DDaCLAE across multiple domains: MNIST for handwritten digit recognition, CIFAR for image recognition, SSTB for sentiment analysis, and SNLI for natural language inference.

6.3.1 Generating Response Patterns

In order to learn the difficulty parameters of the data we require a data set of response patterns. As previously mentioned, gathering enough labels for each example in the data sets to fit an IRT model would be prohibitively expensive for human annotators. In addition, the annotation quality would be suspect due to the humans labeling tens of thousands of examples. Therefore we used artificial crowds to generate our response patterns (Chapter 5).

Briefly, for each data set an ensemble of neural network models is trained, using different subsets of the training data set. Training data is subsampled and corrupted via label flipping so that performance across models in the ensemble is varied. Each trained model then labels all of the examples (train/validation/test). These labels are graded correct/incorrect against the gold-standard label and the output response patterns are used to fit an IRT model for the data (§1.3).

6.3.2 Experiments

In order to demonstrate the effectiveness of DDaCLAE we must show that the model is more efficient than standard supervised learning training while maintaining the level of performance in terms of test set accuracy. Any gains in predictive performance are an additional benefit, but are not the main goal. We will also compare DDaCLAE to a competency-based methods (CB) that uses a fixed, monotonically-increasing competency schedule for adding training examples during training [Platanios

et al., 2019]. For the CB methods below, t is the current time-step in training, T is the point where the model is fully competent, c_0 is the initial competency.

For each data set, we trained a standard model architecture for a set number of epochs. For the NLP tasks we trained a simple LSTM model [Hochreiter and Schmidhuber, 1997]. For SNLI, the model consists of two LSTM sequence-embedding models, one to encode the premise and another to encode the hypothesis. The two sentence encodings are then concatenated and passed through three tanh layers. Finally, the output is passed to a softmax classifier layer to output class probabilities. For SSTB, we used a single LSTM model without the concatenation step. For MNIST we trained a two-layer convolutional neural network (CNN) and for CIFAR we trained a VGG network [Simonyan and Zisserman, 2015, LeCun et al., 2015]. We varied the training data available to the model at each epoch based on the type of curriculum applied:

- Baseline: At each epoch, the model has access to all of the data, shuffled and in mini-batches
- CB-Linear: The proportion of training examples to include at time t is $c_{linear}(t) \triangleq \min(1, t\frac{1-c_0}{T} + c_0)$
- CB-Root: The proportion of training examples to include at time t is $c_{sqrt}(t) \triangleq \min(1, \sqrt{t\frac{1-c_0^2}{T} + c_0^2})$
- DDaCLAE: At each epoch, model ability is estimated ($\hat{\theta}_e$, see §6.2.2) and all training examples where difficulty is less that $\hat{\theta}_e$ are included.

It is worth noting here that neither CB-Linear nor CB-Root actually measures competency of the model at any point. Instead it is assumed that the model becomes more and more competent over time, whereas with DDaCLAE model competency is probed at each training epoch and training data is selected based on this competency.

This is critical because at a given training epoch, there is a chance that less training data is used than the prior epoch, if data was added too quickly.

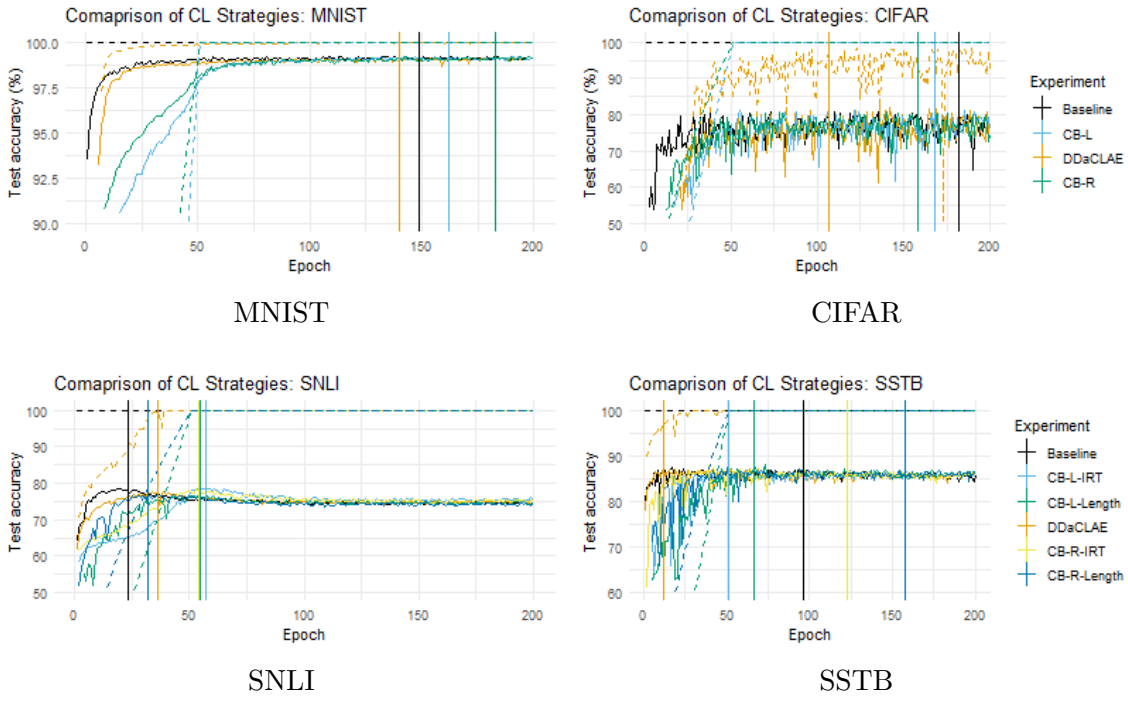


Figure 6.2: Test set accuracy as a function of training epoch for each data set tested. Vertical lines indicate the point at which each method had the highest dev set accuracy (for early stopping). Dotted lines indicate the percentage of training data used by each method at a given epoch. For MNIST, CIFAR, and SSTB, models trained with DDaCLAE converge more quickly than all other training setups. For SNLI, the baseline (training with all data) outperforms all curriculum learning setups. Note: the y-axis has been truncated for each plot to improve visibility. Figure best viewed in color.

Performance in terms of test set accuracy is determined by using the development set accuracy as an early stopping indicator. All models were trained for 200 epochs with development set accuracy used for early stopping.

To determine the effectiveness of difficulty as estimated by IRT methods, we experiment with two versions of the competency-based models in our NLP tasks: (i) d_{length}: using sentence length as a heuristic for difficulty, as in the prior work [Platanios

et al., 2019],[1] (ii) d_{irt}: difficulty as estimated by fitting an IRT model using the artificial crowd (§5.2.4). To the best of our knowledge, most difficulty heuristics for image recognition are based on heuristics from a single model (e.g. confidence score), which we do not consider in this work.

6.4 Results

Using DDaCLAE leads to quicker convergence for the trained models for both the vision and NLP experiments (Figure 6.2). The vertical lines in each plot indicate the point at which the model has converged, based on early stopping using the development set accuracy. For MNIST, CIFAR, and SSTB, DDaCLAE converges more quickly than the baseline models and the competency-schedule baselines. The dotted lines in each figure plot the percentage of training data used for each experiment at a give epoch. For the easier tasks (MNIST and SSTB), DDaCLAE adds training data much more quickly than the competency schedules. Those are artificially holding model learning back due to the rigid structure. What's more, DDaCLAE training data curves are not monotonically increasing (Figure 6.2). If too much training data is added, and model performance suffers as a result (in terms of estimated ability), at the next epoch the hardest examples can be removed until the model is ready for them. This behavior goes against most all other curriculum learning strategies, where more data is added at each epoch.

Along with faster convergence, models trained with DDaCLAE also outperform the other models in terms of test set accuracy (Table 6.1). For SSTB and CIFAR, the models are more efficient (in terms of training data to convergence) and more effective (test set accuracy). In particular, for SSTB training with DDaCLAE leads to 0.45% relative improvement in test set accuracy with an 88.68% relative decrease in training

[1]For NLI, we use the length of the premise sentence to determine difficulty

examples used. Even though training with these curriculum use less data than the baseline, efficiency does not have a significant negative impact on generalizability in terms of test set performance (Table 6.1). The number of training examples required to reach convergence is lower than the baseline in each case except SNLI for DDaCLAE. The baseline SNLI model converges very quickly (after 32 epochs), and because the training data set is so large (550k examples), moving past that point, even slightly, will lead to a large penalty in terms of total number of training examples. For MNIST, the model is more efficient but less effective. We believe this is due to the fact that baseline performance on MNIST is already extremely high (e.g. above 99% accuracy).

We also experimented with using DDaCLAE for hard example mining. At each epoch, only examples where difficulty was greater than $\hat{\theta}_e$, however for all data sets this lead to worse performance. Intuitively there is an upper limit to performance in this case, where the model is trained with only the hardest examples as ability increases, leading to a scenario where the patterns of very difficult examples are learned instead of general class patterns.

By using DDaCLAE a curriculum can adapt during training according to the estimated ability of the model. DDaCLAE adds or removes training data based not on a fixed step schedule but rather by probing the model at each epoch and using the estimated ability to match data to the model (Figure 6.2). This way if a model has a high estimated ability early in training, then more data can be added to the training set more quickly, and learning isn't artificially slowed down due to the curriculum schedule. For each data set in question, DDaCLAE adds training data more quickly than a more traditional curriculum learning schedule, which leads to faster convergence.

Metric	Experiment	MNIST	CIFAR	SSTB	SNLI
%Δ	Baseline	0	0	0	**0**
Train Size	DDaCLAE	**-9.37**	**-53.71**	**-88.68**	33.51
	CB Lin	-8.22	-21.56	-73.17	38.07
	CB Root	11.29	-22.63	10.23	60.08
%Δ	Baseline	**0**	0	0	0
Accuracy	DDaCLAE	-0.17	**0.66**	**0.45**	-1.08
	CB Lin	-0.01	-0.90	-0.18	**0.69**
	CB Root	-0.06	0.13	-0.38	-0.37

Table 6.1: Percent change in training size (lower is better) and test set accuracy (higher is better) for each curriculum learning method tested.

6.4.1 Discrepancies in difficulty

The differences in results between the methods that use d_{length} and d_{irt} for NLP indicate that there are cases where learned difficulty does not match the expectations of human heuristics. We calculated the absolute difference between difficulties to identify those with the largest discrepancies for further analysis: $\Delta_d = |d_{irt} - d_{length}|$ The cases where discrepancy is largest indicate clear patterns that are worth discussing (Tables 6.2 and 6.3).

For sentiment analysis, there are a number of easy to classify examples that are very long. The authors of these review snippets are clear in their like/dislike for a particular film, but use a lot of words to make their point. These are interesting examples because they could be very useful for a model early in training to identify patterns of long-but-easy reviews. On the other hand, there are one-word snippets that are very difficult to classify and would cause problems for a model if introduced very early during training.

For NLI, some of the biggest discrepancies come from short, neutral examples. Models find these very difficult because there is not enough information to determine that the premise neither entails nor contradicts the hypothesis. On the other hand,

examples with long premises are easier to deal with because (a) there is a short but clear contradiction in the hypothesis or (b) the hypothesis is neutral and irrelevant.

Label	Review	Δ_d
Pos	Heart	67342
Pos	Leaping from one arresting image to another, Songs from the Second Floor has all the enjoyable randomness of a very lively dream and so manages to be compelling, amusing and unsettling at the same time.	67339
Pos	The year's greatest adventure, and Jackson's limited but enthusiastic adaptation has made literature literal without killing its soul – a feat any thinking person is bound to appreciate.	67334
Pos	Hip	67332
Neg	Exit	67346
Neg	In theory, a middle-aged romance pairing Clayburgh and Tambor sounds promising, but in practice it's something else altogether – clownish and offensive and nothing at all like real life.	67337
Neg	There's an admirable rigor to Jimmy's relentless anger, and to the script's refusal of a happy ending, but as those monologues stretch on and on, you realize there's no place for this story to go but down.	67330

Table 6.2: Examples from SSTB with the largest differences in difficulty.

Label	Premise	Hypothesis	Δ_d
Cont.	Two men in a jogging race on a black top street, one man wearing a black top and pants and the other is dressed as a nun with bright red tennis shoes, while onlookers stand in a grassy area and watch from behind a waist high metal railing.	There is no metal railing.	549179
Ent.	Two dogs in the water.	They are swimming	549180
Neut.	Male musicians are playing a gig with one on the drums and the other on the guitar, with a backdrop of purple graphics apart of the light show.	Male musicians with long hair are playing a gig with one on the drums and the other on the guitar, with a backdrop of purple graphics apart of the light show.	549184
Neut.	A dog in a lake.	A dog is swimming.	549183
Neut.	A man rock-climbing	The man is outdoors.	549181

Table 6.3: Examples from SSTB with the largest differences in difficulty.

6.5 Conclusion

DDaCLAE is the first curriculum learning method to dynamically probe a model during training to estimate model ability at a point in time. Knowing the model's ability allows for data to be selected for training that is appropriate for the model and is not rigidly tied to a heuristic schedule. Learning with DDaCLAE curriculum learning strategies leads to more efficient models. The difference in performance is small for many tasks, but being able to reduce training data by 50% or more allows for the use of certain models in many more cases, and also allows for more researchers to work on problems where huge computing and storage resources may not be available.

A key component of most prior work in curriculum learning is the notion of balance. When defining a curriculum, it is often the case that proportions are maintained between classes. That is, difficulty itself is not the only factor when building the curriculum. Instead, the easiest examples for each class are added so that the model is proportionally exposed to the data consistent with the full training set. DDaCLAE does not consider class labels when selecting examples for training.[2] In this way DDaCLAE is more closely aligned with a pure curriculum learning strategy that considers only the easiness/hardness of an example during training. This is an added benefit to the method as there is no need for class label accounting during training.

Even though it is dynamic, DDaCLAE employs a simple curriculum schedule: only include examples where difficulty is less than estimated ability. However, being able to estimate ability on the fly with DDaCLAE opens up as a research area the following: what is the best way to build a curriculum, knowing example difficulty and model ability? It may be the case that only data with difficulty within a range of ability (higher and lower) is better, and the training set shifts as the model improves.

[2]It is important to note here that labels are used when learning difficulties, estimating ability, and actually updating parameters during training. They are not used to balance the curriculum.

There are many directions to for future work, and this will be an exciting area of work moving forward.

CHAPTER 7

COMPREHENOTES: ASSESSING PATIENT READING COMPREHENSION OF ELECTRONIC HEALTH RECORD NOTES

7.1 Introduction

Providing patients access to their medical records through personal health records (PHRs) is becoming more common as physicians move to electronic health record (EHR) systems. PHRs are defined as "electronic, lifelong resource of health information needed by individuals to make health decisions" [Burrington-Brown et al., 2005]. Providing patients direct access to their EHR clinical notes can enhance patients' understanding of their clinical conditions and improve their health care outcomes [Ross and Lin, 2003,Honeyman et al., 2005,Delbanco et al., 2012]. For example, the Veterans Health Administration offers the MyHealth*e*Vet PHR through a Web-based patient portal, which allows millions of veterans to view their EHRs [Nazi et al., 2013]. These records include both structured (e.g., patient vitals) and unstructured data (e.g., discharge summaries and clinical notes). However, patients with limited health literacy may struggle to understand the content of their medical notes, which can include visit summaries with medical terms, lab reports, and terms and phrases that are not common outside of medicine. A patient's health literacy can have an impact on their desire to engage with their own PHR [Noblin et al., 2012,Irizarry et al., 2015].

Low health literacy can impact a patient's ability to communicate with their health care providers and to navigate and understand complex EHR information. Health literacy is defined by the Institute of Medicine as "the degree to which individuals have the capacity to obtain, process, and understand basic information and services

needed to make appropriate decisions regarding their health" [Lynn Nielsen-Bohlman, 2004]. According to the National Assessment of Adult Literacy, only 12% of adults are proficient in health literacy [Kutner et al., 2006]. The average American reads at or below an eighth grade level, and over 90 million Americans have limited health literacy [Kutner et al., 2006]. Moreover, 50% of patients do not understand at least one term in their medical problem list [Lynn Nielsen-Bohlman, 2004, Jones et al., 1992, Lober et al., 2006]. In addition, EHR notes do not align well with existing readability prediction formulas, making it difficult to estimate EHR note readability [Zheng and Yu, 2017]. Consider the following example, taken from a de-identified EHR clinical note: "The monitor has not shown any dysrhythmias or arrhythmia either prior to or during any of his spells." A patient might struggle to understand the medical terms dysrhythmias and arrhythmia and might not understand what the monitor is or what prior to or during any of his spells is referring to.

Low health literacy can lead to serious problems. For example, low health literacy was shown to be independently associated with an increase in mortality among the elderly [Sudore et al., 2006]. A recent assessment of health literacy involving over 400 Veterans found that 87% of Veterans have low health literacy [Schapira et al., 2012]. Most health care consumers do not understand phrases often used in cancer consultations [Chapman et al., 2003]. Patients understand less than 30% of medical terms commonly used in the emergency department [Lerner et al., 2000]. Patients with low health literacy are more likely to lack awareness of their atrial fibrillation diagnosis [Reading et al., 2017] and are at higher risk for increased fear of cancer progression [Halbach et al., 2016].

Given the prevalence of low health literacy in the population, tools that effectively assess a patient's health literacy are needed for both research and practice. Of the existing instruments, 3 that are widely used are the Rapid Estimate of Adult Literacy in Medicine (REALM), the Test of Functional Health Literacy in Adults (TOFHLA),

and the Newest Vital Sign (NVS) [Davis et al., 1993, Parker et al., 1995, Weiss et al., 2005]. Each of these has value, but also limitations. For example, REALM can be administered in 2 to 3 min, but it assesses word recognition, not comprehension [Davis et al., 1993]. TOFHLA assesses reading comprehension and numeracy using passages from health care-related documents, hospital forms, and prescription labels [Parker et al., 1995]; a short version of TOFHLA reduced the administration time from 22 min to 12 min [Baker et al., 1999]. NVS contains 6 items tied to a single stimulus (a food label) and can be administered in 3 min. It was intended as a screening tool and is less appropriate for generating scores that discriminate between different levels of health literacy in patients [Weiss et al., 2005, Osborn et al., 2007]. Taken together, these tests can provide information on a patient's general health literacy, but none assesses a patient's ability to comprehend EHR notes.

The purpose of this study was to create an instrument to measure EHR note comprehension in patients. We first identified a set of representative EHR notes for 6 diseases and conditions from a large hospital EHR system. From these notes, a group of physicians and medical researchers generated questions using the Sentence Verification Technique (SVT) [Royer et al., 1979, Mazor et al., 2012b, Mazor et al., 2012a]. We obtained responses for these questions from the crowdsourcing platform Amazon Mechanical Turk (AMT) and analyzed the results using Item Response Theory (IRT) [Baker and Kim, 2004, Fries et al., 2005, Nguyen et al., 2016, Diviani et al., 2017] to select a subset of questions for a test of EHR note comprehension.[1] To the best of our knowledge, the ComprehENotes question set is the first instrument to assess EHR note comprehension.

The goal of this work is to develop a set of questions that can be used to test patient EHR note comprehension. To that end we developed a process for note selection,

[1]For this chapter, because we are discussing a more traditional application of IRT, we return to the traditional IRT terminology and refer to items as "items."

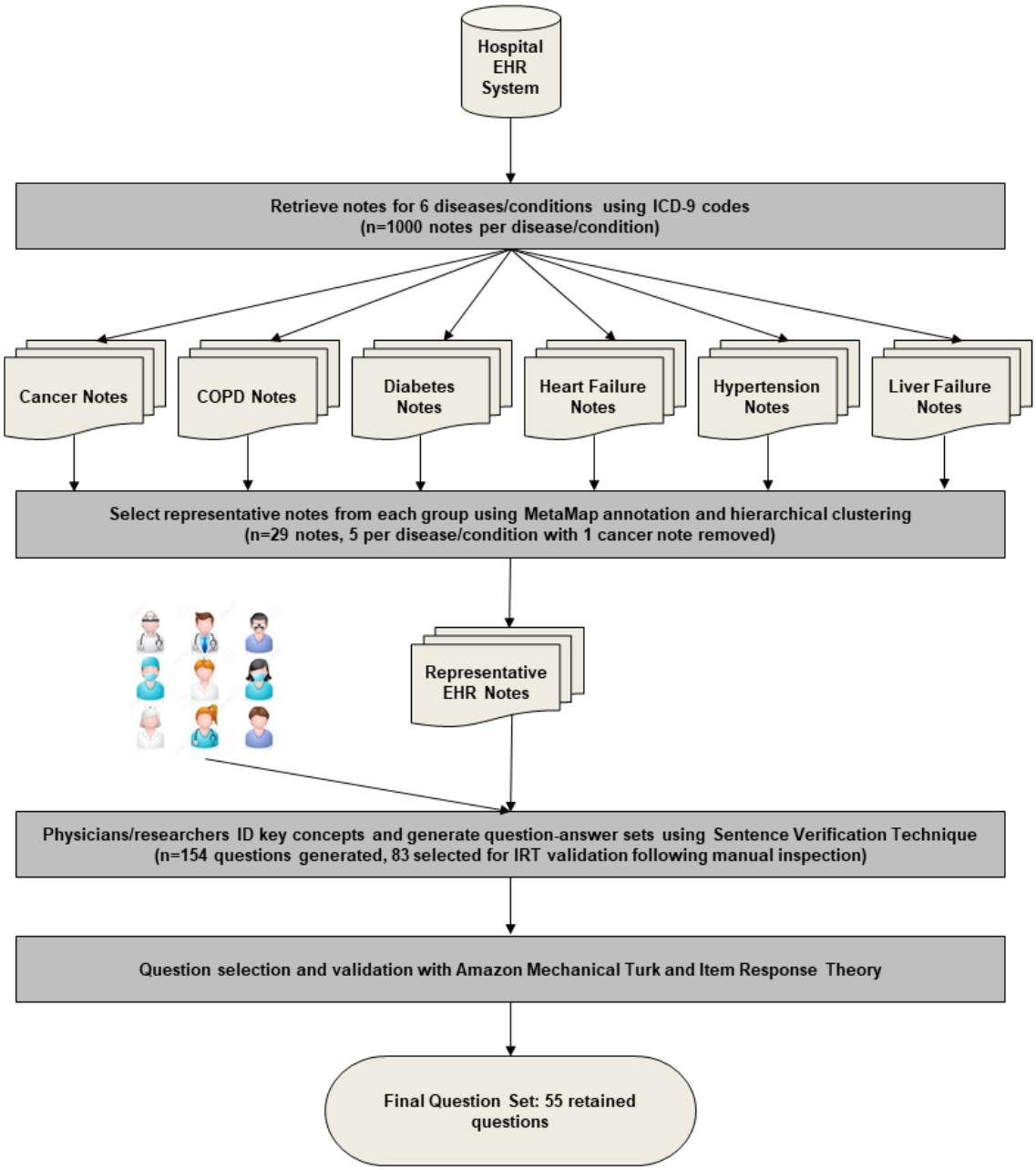


Figure 7.1: Visualization of the question generation and validation process for the ComprehENotes test set.

question generation, and question selection and validation (Figure 7.1). We discuss each step in detail in the following sections. The first step is to identify a candidate pool of questions that can ultimately be filtered down using IRT to become a test set. To this end we use a topic model to identify representative EHR notes that were given to a set of medical professionals to write questions. Once these questions were generated and reviewed, we again used AMT to get enough response patterns to fit our IRT model and generate a test set. To account for patients who may not have a lot of time to take the test, both a full test and a short-form that can be easily administered at a hospital as part of a check-up are presented.

7.2 Building ComprehENotes

7.2.1 EHR Note Selection

We selected notes according to International Classification of Disease (ICD-9) codes associated with six important and common diseases: heart failure (428), hypertension (401), diabetes (249, 250), COPD (493.2, 491, 492, 494, 496, 506), liver failure (571), and cancer (140-239). By selecting notes from multiple diseases our goal was to obtain a variety of notes associated with common diseases in order to generate questions across multiple topics. We retrieved EHR discharge summary and progress notes from the University of Massachusetts Memorial Hospital EHR system. Progress notes provide information regarding a patient's conditions and treatments, while discharge notes may include a summary of the patient's visit, necessary patient follow-up, and other information. These types of notes include information that is relevant to patients and are good candidates for question generation. For each disease/condition we randomly selected 1000 notes. Since the EHR notes vary significantly for length (anywhere between 50 words to over 1500 words), we limited the note selection to notes between 300 and 1000 words long. Notes that are longer than 1000 words often contain duplicate information or large tables of lab results with few free-text section from which we can

generate questions. We annotated each note with MetaMap [Aronson and Lang, 2010] to map the note to Unified Medical Language System (UMLS) concepts [Bodenreider, 2004]. For each category, we ran topic modeling on the 1000 notes using the UMLS concepts that were identified by MetaMap and hierarchically clustered the notes into 5 clusters based on topic similarities. Finally, we selected one representative note (the note with the most UMLS concepts) from each cluster. By selecting the note with the most concepts our goal was to identify those notes with the most information that could be used as part of the question generation process. This procedure resulted in a total of 30 notes, with 5 notes per disease/symptom. We discarded one cancer note because the physicians identified it as a pure lab test report that did not include any natural language text.

7.2.2 Generating Questions with SVT

SVT is a procedure for generating reading comprehension items to evaluate whether an individual has understood a passage of text [Royer et al., 1979, Royer et al., 1987, Royer, 2004]. SVT has been applied in many different reading comprehension environments, such as basic language research [Kardash et al., 1988], evaluating the effect of prior beliefs on comprehension [Kardash and Scholes, 1995], and assessing language skills of non-native English speakers [Royer and Carlo, 1991]. In addition, SVT has been used to develop tests to assess comprehension of cancer screening and prevention messages [Mazor et al., 2012b, Mazor et al., 2012a]. SVT tests are sensitive to both differences in reading skill and text difficulty. Tests using SVT questions have been shown to be effective for measuring reading comprehension, and for assessing comprehension of written and spoken health messages [Mazor et al., 2012b, Mazor et al., 2012a].

We asked experts to create question-answer sets following two steps: (1) identifying important content in the notes, and (2) creating comprehension test questions.

Specifically, the selected 29 de-identified notes were provided to 5 groups. Each group included one physician and 2-3 non-clinician researchers (a total of 4 physicians and 13 researchers where one physician participated in two groups). The groups were given an introduction to the SVT methodology before taking part in the exercise. Each member read every assigned EHR note and then identified important content (usually a sentence). Each member then followed SVT protocol to create question-answer sets for the identified content.

An SVT test is designed by taking a sentence or phrase from a passage of text (the "original") and generating three additional sentences or phrases: (i) a "paraphrase" where as much of the sentence/phrase is changed as possible while preserving the original meaning, (ii) a "meaning change" where the original sentence/phrase is changed slightly, but enough that the original meaning is changed, and (iii) a "distractor" that is unrelated to the original but still consistent with the passage theme [Royer et al., 1979].

Once generated the question-answer sets were then discussed in the group and a final question-answer set was agreed upon. 154 question-answer sets were generated from the 29 EHR notes. Table 7.1 shows an example of a question-answer set generated by the groups, and Table 7.2 shows how this question would be presented to patients in a test scenario. We selected 83 of the 154 questions for further analysis. Questions were selected based on their content. We manually selected questions that were generally relevant to the main topic (e.g. diabetes) over questions that were very specific to a patient's note to keep the question set general enough to be given to future patients. 11 to 13 question-answer sets were retained for 4 of the 6 topics, and 18 question-answer sets were retained for COPD and Diabetes.

EHR Note Text	The monitor has not shown any dysrhythmias or arrhythmia either prior to or during any of his spells
Paraphrase	His heart rhythm is normal before and during his fainting spells
Meaning Change	He has had abnormal rhythm prior to or during his spells of chest pain
Distractor	The monitor has shown abnormal heart rhythms before and during his spells

Table 7.1: Example of questions generated from the researcher/physician groups.

Please read the following question and then examine the answer choices and choose the answer that best represents the question text. What does the following sentence mean? "The monitor has not shown any dysrhythmias or arrhythmia either prior to or during any of his spells." 1. He has had abnormal rhythm prior to or during his spells of chest pain 2. The monitor has shown abnormal heart rhythms before and during his spells 3. His heart rhythm is normal before and during his fainting spells

Table 7.2: Examples of how the generated questions would be displayed as a questionnaire, using the example from Table 7.1.

7.2.3 Data Collection

To gather enough human responses to fit the IRT model, we recruited participants from AMT. We created 6 comprehension tasks on AMT, one per disease topic, to analyze each topic separately. Each task was completed by 250 AMT workers (Turkers), who were presented with the test questions, one question at a time. This sample size is large enough to satisfy the accepted standards for IRT models based on the non-central distribution [MacCallum et al., 1996]. We collected demographic information from the Turkers prior to administering the test questions, and we implemented several quality control mechanisms to ensure the quality of the Turker results. Only Turkers with approval rates above 95% and located in the United States were able to participate. The 95% approval rate identifies Turkers that have been approved most of the time according to their completion of other tasks on AMT and is indicative of the high

quality of their previous tasks. Restricting the task to users located in the United States is used as a proxy for English proficiency. In addition, in each test one question was randomly selected as a quality-check question and was presented to the Turker twice during the course of the evaluation. If the Turker gave two different answers to the repeated question his responses were not included in later analyses. Two simple questions were also added to the test as quality control. If the Turker answered one or both of the quality control questions incorrectly his responses were rejected from consideration and not included in later analyses.

For the COPD and Diabetes tests, the 18 questions were split into 3 groups of 6 questions. Each Turker was given a random selection of two of the three groups. This way the test lengths were similar to the other disease topic tests, and the conditions in which Turkers provided responses were consistent across the groups. For the COPD and Diabetes tasks we recruited 400 Turkers so that the number of responses per question were consistent with the other topics.

7.2.4 Item Analysis and Selection using Item Response Theory

After data collection, the Turker responses were analyzed using a 3-parameter logistic (3PL) IRT model.

The 3PL model was fit to data for each set of questions using the R packages mirt and ltm [Rizopoulos, 2006, Chalmers et al., 2012]. Marginal residuals of each pair of items and each triplet of items were checked and items that gave large residuals were removed for violation of local independence. Items with a negative slope were also removed. Guessing parameters not significantly different from zero were set to zero. A key parameter used to identify a "good" question for future evaluations is the slope of the item characteristic curve. If the slope is flat, then the item cannot distinguish between individuals of high ability levels and individuals of low ability levels. After refitting the remaining items, items with a slope parameter not significantly greater

than zero or less than 0.71 were removed. The value 0.71 corresponds to a communality of 0.15 in an exploratory factor analysis, which means that 15% of the variance of the item would be explained by the latent ability factor if the item were continuous. 55 items were retained in this analysis for further validation. From the 55 items we also identified 14 of the 55 items with the largest slopes (discrimination parameters) and highest average information for inclusion in the short form of the test. The short test should be as informative as possible while reducing the length of the test, making it more practical to administer.

7.2.5 Confirmatory Evaluation of Item Quality

The questions retained from the initial IRT analysis were combined into a single test and deployed in a new AMT task to validate the item parameters. For this task, we split the 55 retained questions into 3 groups (each of 18-19 questions) and created 3 AMT tasks where Turkers were shown 2 of the 3 groups and asked for responses as above. Quality checks were included as in the first set of AMT tasks. For these tasks Turkers that participated in the initial data collection were excluded. Responses were generated and a second round of IRT analysis was performed to confirm that the questions retained from the first round could be considered a cohesive test of EHR note comprehension as a whole.

7.2.6 AMT Responses and Turker Demographics

We first report descriptive statistics and demographic information about the Turkers who completed the per-topic and validation AMT tasks (Figure 7.2, Table 2). Responses for both the per-topic and validation tasks covered a wide range of correctly answered questions, with mean scores for each task above 70%. Across all tasks no more than 10% of responses were removed because of quality control checks.

We also looked at raw scores and estimated ability in the validation task to see if there were patterns in the responses that matched expected behavior (Table 3).

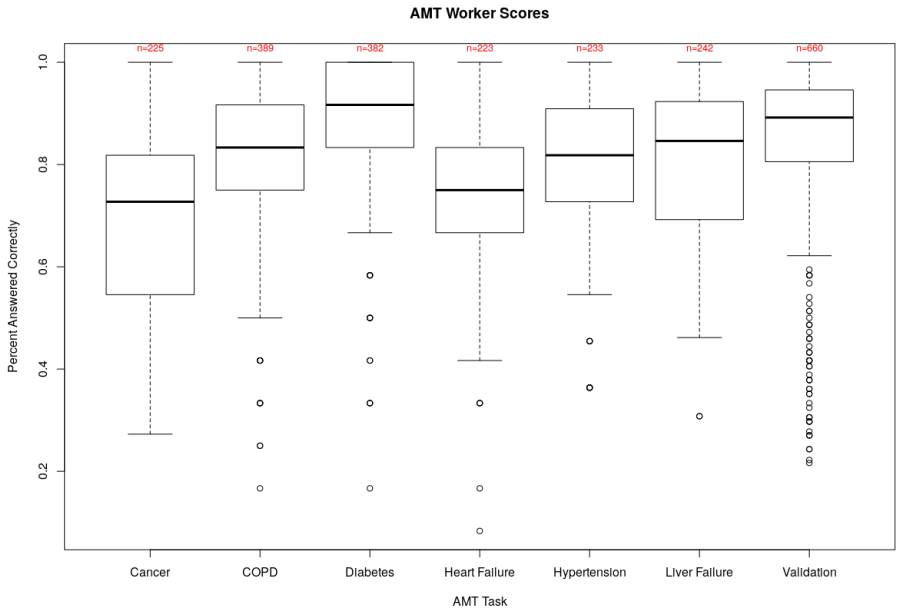


Figure 7.2: Box plots of Turker scores on the AMT per-topic and validation tasks. Average raw score is above 70% in all cases. Counts indicate the number of AMT responses retained after quality-control.

As expected, mean scores for individuals with more education are higher than for individuals with less education. In addition, Turkers over 45 score higher on average than Turkers under 45. There is a slight drop in mean scores for Turkers over 65 which makes sense given that adults ages 65 and older have lower health literacy on average [Kutner et al., 2006].

7.2.7 Item Selection

55 of the 83 questions (66%) provided to Turkers in the per-topic AMT tasks were retained after the initial IRT analysis (Figure 7.3). Items were identified for removal according to the procedure identified in the Methods section. Table 4 shows examples of retained and removed items. In the case of the removed item, the question simply defining the term "Osteoporosis" was too easy for the Turker population. That is, most of the Turkers answered the question correctly, so the probability of answering the question correctly is very high even at low levels of ability. A question like this

Demographic	Value	**Per-topic count** n (%), (n=1694)	**Validation count** n (%), (n=664)
Gender	Male	880 (51.9)	253 (38.1)
	Female	814 (48.1)	411 (61.9)
Race	African American	107 (6.3)	59 (8.9)
	Asian	163 (9.6)	51 (7.7)
	Hispanic	89 (5.3)	32 (4.8)
	American Indian	7 (0.4)	12 (1.8)
	Pacific Islander	9 (0.5)	0 (0)
	White	1319 (77.9)	510 (76.8)
Education	Less than High School	17 (1.0)	4 (0.6)
	High School Degree	504 (29.8)	189 (28.5)
	Associate's Degree	283 (16.7)	109 (16.4)
	Bachelor's Degree	697 (41.1)	256 (38.6)
	Master's Degree or Higher	193 (11.4)	106 (16.0)
Age[a]	18-21	n/a	14 (2.1)
	22-34	n/a	331 (49.8)
	35-44	n/a	158 (23.8)
	45-54	n/a	106 (16.0)
	55-64	n/a	40 (6.0)
	65 and older	n/a	15 (2.3)

Table 7.3: Demographic information of Turkers from the per-topic and validation AMT tasks. [a]Age demographic information was not collected as part of the per-topic AMT tasks.

does not give us any information about an individual's ability and therefore is not needed in the test set.

The test information curve is presented in Figure 7.4. Test information is defined as the reciprocal of the squared Standard Error (SE) of the ability estimate: where σ is the SE [Baker and Kim, 2004]. Test information measures how accurate the ability estimates are at varying levels of ability. Given that most items have negative difficulty, the information curve has high values in the negative ability levels. That is, estimates of ability for negative ability levels are more accurate. Test information is greater than 4 for the range of ability levels between -2.8 and 0.7, which means for this range of ability levels (from 2.8 SDs below to 0.7 SD above the average of

Demographic Characteristic	Mean Percent Correct (%)	Average Estimated Ability
Education		
Less than High School	64.7	0.899
High School Degree	84.9	0.038
Associate's Degree	83.8	0.013
Bachelor's Degree	83.8	0.034
Master's Degree or Higher	88.1	0.199
Age		
18-21	77.4	0.493
22-34	83.7	0.042
35-44	83.6	0.066
45-54	88.3	0.222
55-64	89.4	0.212
65 and older	85.9	0.122
Gender		
Male	80.6	0.236
Female	87.2	0.143

Table 7.4: Average estimated ability of Turkers according to demographic information for the validation task.

the population of AMT users), the SE of an ability estimate is smaller than 0.5. The full test is most informative in ability around -2 with maximum information of 44.2 (Figure 5, red dotted line). This maximum is mostly due to a single item (44) with the largest slope of 11.3. Because of the very large slope parameter, this item is very informative around ability of -2, but is not informative at other areas of ability. Since one goal of the test is to identify individuals with low ability, this item may be useful and is therefore included in our test set. However, we also wanted to confirm that the other test questions are still informative in their own right. To do this we plotted the test information curve without item 44. Without this item, the item information curve is most informative around -1.5, with a maximum of 30.6 (Figure 5, black solid line).

The test information curve of this short test is also presented in Figure 7.4. The short test includes item 44, so we also plot information for a 13-item test without item 44. For the short test, test information is greater than 4 (i.e. SE of ability estimate is

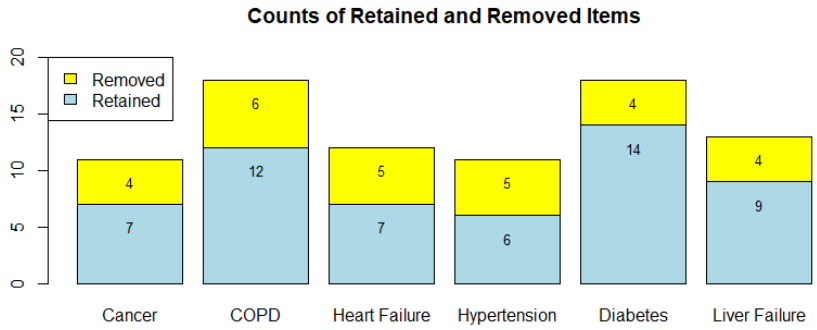


Figure 7.3: Results of analysis to identify useful items from the question sets. Items were removed according to the reasons outlined in the Methodology.

Retained	Question	Pegfilgrastim 6 mg subcutaneous one dose
	Paraphrase	Do an under skin injection of one dose of 6 mg pegfilgrastim
	Meaning change	Pegfilgrastim 6 mg epidermal one dose
	Distractor	Pegfilgrastim may prevent neutropenia
Removed	Question	Osteoporosis
	Paraphrase	Weakness in bones
	Meaning change	Hardening of bones as we get older
	Distractor	Some bones get hard and some weak

Table 7.5: Examples of retained and removed questions following IRT analysis.

smaller than 0.5) in the range between -2.4 and -0.5, or 2.4 SDs to 0.5 SD below the average AMT user, again appropriate for a population of low literacy.

7.3 Validation with an Education Intervention

In recent years, many hospitals have adopted patient portals to make medical records available to patients. In particular, patient portals allow patients to access their electronic health records (EHRs). In a survey of studies related to patient access to their medical records, generally, patients who chose to see their records were satisfied with their contents [Ross and Lin, 2003, Masys et al., 2002, Sheldon, 1982, Bronson et al., 1986] and felt greater autonomy about their care [Ross and Lin, 2003, Homer et al., 1999, Draper et al., 1986]. Granting patients access to their records also does

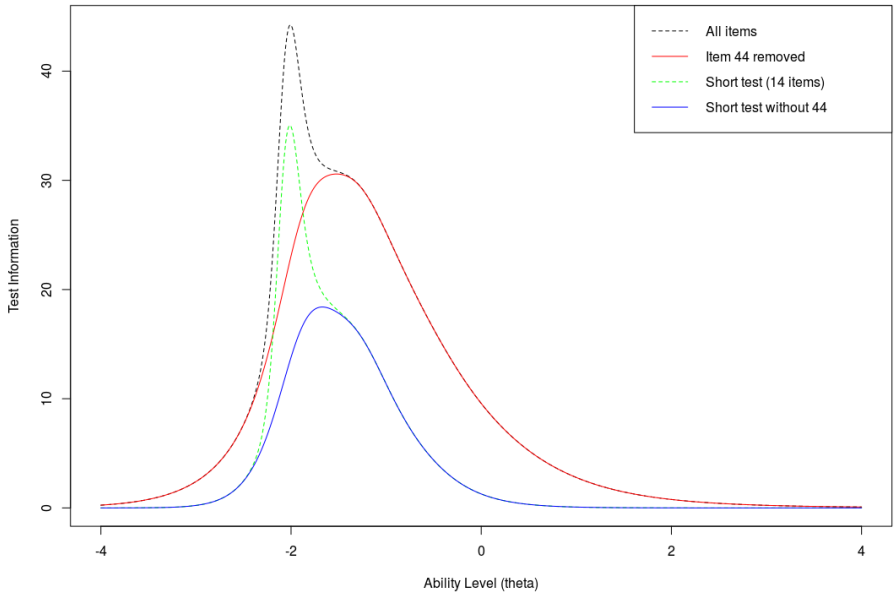


Figure 7.4: Test information curve for the full ComprehENotes instrument (55 items) and various subsets.

not increase the workload of medical staff members [Ross and Lin, 2003, Hertz et al., 1976, Baldry et al., 1986, Golodetz et al., 1976]. Generally, patient access to EHRs can lead to positive health outcomes and greater understanding of their conditions [Ross and Lin, 2003, Honeyman et al., 2005, Delbanco et al., 2012]. However, EHRs and the progress notes that are included often contain complex medical jargon that is difficult for patients to comprehend. When given access to their notes, patients have questions about the meaning of medical terms and other concepts included in the notes [Golodetz et al., 1976, Jones et al., 1992]. Tools such as OpenNotes have promoted the inclusion of patient visit notes in patient portals, but simply including the notes may not be beneficial for patients if they have questions regarding the meaning of terms in the notes. Tools and resources that can define terms and provide lay definitions for medical concepts are needed as part of the move to make EHR notes available to patients so that they can understand the contents of their notes and their medical record.

Self-service educational materials are widely available, especially on the Web. There is a wealth of information related to medicine and health care on the internet, ranging from well-maintained ontologies with curated educational materials to Web-based discussion communities of patients that suffer from the same disease. With this information, patients with certain symptoms can find information about their condition on the internet. But is the wealth of information useful? That is, does simply having access to health information lead to better understanding? We test the usefulness of both passive and active interventions for assisting patients with understanding medical concepts. The passive system, MedlinePlus (MLP) [Miller et al., 2000], is a Web-based repository maintained by the US National Library of Medicine that includes information and definitions for clinical concepts, diseases, and other terms related to health care. MLP has been used in the past to promote patient education and provide patients with definitions and educational material to improve health literacy [Coberly et al., 2010, Gaines et al., 2011, McMullen et al., 2011, Teolis, 2010]. MLP is a large repository of high-quality health care information, but the user must search for the information that he or she is looking for. MLP does not automatically surface information for users.

NoteAid [Polepalli et al., 2013] is a freely available Web-based system developed by our team that automatically identifies medical concepts and displays their definitions to users. NoteAid has previously been shown to improve patients' understanding of notes as measured by self-reporting [Polepalli et al., 2013].

Our goal is to determine if access to NoteAid or MLP is associated with higher levels of EHR note comprehension. Do these interventions of educational materials improve a patient's ability to comprehend his or her EHR note? We use the Amazon Mechanical Turk (AMT) microtask crowdsourcing platform to give AMT workers (Turkers) the ComprehENotes EHR note comprehension test, a set of questions designed to test EHR note comprehension. AMT is an increasingly popular tool for gathering research

data [Snow et al., 2008, Sabou et al., 2012] and recruiting participants for experiments, both in open-domain tasks [Demartini et al., 2012] and medical-specific research [Zhai et al., 2013, Good et al., 2015, Mortensen et al., 2015, Gottlieb et al., 2015]. Certain Turkers were not given 1 of the external resources, whereas others were provided with either MLP or NoteAid. Our results show that using NoteAid leads to significantly higher scores on the EHR comprehension test compared with the baseline population that was given no external resource. However, we found no significant difference between the Turkers with no resource and the Turkers who used MLP. Turkers were also asked to take the short Test of Functional Health Literacy in Adults (S-TOFHLA) to assess functional health literacy. All the Turkers scored adequate health literacy, the highest level for S-TOFHLA. This is the first work to quantitatively analyze the impact of tools such as NoteAid using a test of EHR note comprehension as opposed to self-reported scores.

We show that NoteAid has a significant impact on EHR note comprehension as measured by a test specific to that task. In addition, simply giving a patient access to sites such as MLP does not lead to significant improvements in test scores over a baseline group that had no external resources available to them. Finally, we analyze the demographics of the Turkers who completed our tasks. A regression model to predict test scores showed differences between demographic groups that align with the current knowledge regarding health literacy. For example, individuals that reported education of less than high school scored lower than average, whereas individuals that identified as white scored higher than average.

Health literacy is an important issue for patients. Low health literacy is a widespread problem, with only 12% of adults estimated to be proficient in health literacy [Kutner et al., 2006]. The Institute of Medicine defines health literacy as "the degree to which individuals have the capacity to make appropriate decisions regarding their health" [Lynn Nielsen-Bohlman, 2004]. Patients with low health literacy often

have difficulty with understanding instructions for medications from their doctors and have trouble navigating systems for making appointments, filling prescriptions, and fulfilling other health-related tasks [Lerner et al., 2000, Chapman et al., 2003]. In addition, having low health literacy has been linked to negative health outcomes in areas such as heart disease and fear of cancer progression [Halbach et al., 2016, Reading et al., 2017].

It is important to be able to test a patient's health literacy to identify those patients with low health literacy. Doctors can then provide these patients with educational materials to improve their understanding of medical terms and concepts. Testing health literacy is especially important with the proliferation of Web-based patient portals, where patients can access their EHRs and EHR notes directly. Giving a patient access to their EHRs and EHR notes without confirming that the patient can understand the content of the notes may lead to confusion and frustration with their health care experience.

There are a number of tests for health literacy, including the Test of Functional Health Literacy in Adults (TOFHLA) and the Newest Vital Sign (NVS) [Parker et al., 1995, Baker et al., 1999, Weiss et al., 2005]. TOFHLA and its shortened form (S-TOFHLA) test comprehension and numeracy by providing scenarios to patients and constructing fill-in-the-blank questions by removing key terms from the scenario passages. NVS is a short test where patients are required to answer questions related to a nutrition label, to test whether the patient can navigate the label. These tests work well as screening instruments to identify patients who may have low health literacy, but they are broad tests and do not specifically test EHR note comprehension.

Although these and other tests are available, the only test that specifically targets a patient's ability to comprehend their EHR notes is the ComprehENotes test. The ComprehENotes test questions were developed using key concepts extracted from de-identified EHR notes. Questions were written by physicians and medical researchers

using Sentence Verification Technique and validated using Item Response Theory (IRT) [Royer et al., 1979, Baker and Kim, 2004]. The test set is the first of its kind that specifically tests a patient's ability to comprehend the type of content that is included in EHR notes.

7.3.1 Methods Overview

We recruited Turkers on the AMT platform and asked them to complete the ComprehENotes EHR note comprehension test. Turkers were split into 3 groups and were allowed to use 1 external resource when completing the test (or no resource in the case of the baseline group). Test results were collected and analyzed using IRT to estimate EHR note comprehension ability for each of the individuals, and group results were analyzed to determine if either of the external resources had a significant effect on test scores. Figure 7.5 illustrates our methodology at a high level. Details for each of the steps are described below.

7.3.2 Data Collection

The questions in the ComprehENotes test set include questions from patient EHR notes associated with 6 diseases: heart failure, hypertension, diabetes, chronic obstructive pulmonary disease (COPD), liver failure, and cancer. The questions are all general enough that they assess a key concept associated with 1 of the 6 diseases without being so specific to a single patient that they are not useful to others. Therefore, the test can be used to assess a patient's general EHR note comprehension ability and allows for comparisons between patients with respect to comprehension ability.

The ComprehENotes test set is most informative for individuals with low health literacy. That is, the SE of the ability estimation is lowest at low levels of ability (e.g., -2 to -0.5). In addition, most of the ComprehENotes questions have low difficulty parameters. The difficulty parameters range from -2.2 to 0.7. That is, the questions

are of a difficulty that individuals with lower than average ability have a 50% chance of answering correctly. For example, if a question has a difficulty parameter of -1.0, then an individual with estimated ability of -1.0 has a 50% chance of answering the question correctly. Ability estimates are normally distributed, so an individual with estimated ability of -1.0 is 1 SD below the average individual. Individuals are shown a snippet of text from a de-identified EHR note and asked to select the answer that has the same meaning as the italicized portion of the text.

We set up 3 AMT tasks for Turkers to complete. Turkers were presented with the ComprehENotes question set, 1 question at a time, and were asked to provide the correct answer.

For 1 task (Baseline), the Turkers were instructed to not use any external resources when answering the questions. For the first treatment task (Treatment-MLP), Turkers were given a link to MLP and were told that they could use the site as a reference when completing the task. Turkers were encouraged to use the MLP page search functionality to search for definitions to unknown terms or concepts that appeared in the task. For the second treatment task (Treatment-NoteAid [Treatment-NA]), the Turkers were provided with a version of the ComprehENotes test set that had been preprocessed with NoteAid. We preprocessed the ComprehENotes question text using NoteAid, extracted the simplifications and definitions that were provided, and used the NoteAid output as the question text shown to Turkers in the Treatment-NA group (refer to Figure 7.6 for an example of text simplified by NoteAid). The tasks were restricted so that individuals who completed 1 were not eligible to complete the other 2. For all groups, we collected demographic information about the Turkers' age, gender, ethnicity, level of education, and occupation. We also administered the S-TOFHLA test for each group to assess functional health literacy and to compare S-TOFHLA and ComprehENotes scores.

As we are not able to monitor the Turkers as they complete our tasks, we cannot know for sure that the baseline group did not use any external resources as instructed. However, we can be confident that they did not have access to NoteAid. To access NoteAid, the Turkers would have to have known the URL link to access the system, even though we did not provide it to them. Alternatively, the Turkers would have had to search for NoteAid without knowing the name of the specific system we are testing. Therefore, we are confident that even if the baseline group did use some external source during the task, they did not have access to NoteAid. The baseline Turkers may have found MLP if they searched on the Web for medical concepts during the task. For example, a Google search of "COPD definition" returns an MLP link on the first page. However, unless the Turkers knew about MLP before beginning the task, it is unlikely that they would use MLP as a reference during the task.

We included quality control checks for our AMT tasks to ensure a high-quality response from the Turkers. First, we restricted access to our tasks to Turkers with a prior approval rating above 95% to include only Turkers whose work has been judged as high quality by other requesters. We also restricted the task to Turkers located in the United States as a proxy for a test of English proficiency. Within the actual task, we included 3 quality-check questions, which consisted of a very simple question with an obvious answer. If any Turker answered 1 or more of the quality control checks incorrectly, their responses were removed from the later analyses.

The NoteAid system supplies lay definitions for medical concepts in EHR notes [Polepalli et al., 2013]. Users enter the text from their EHR notes into the NoteAid system, which outputs a version of the note with medical concepts defined. When the user hovers his or her mouse over a concept, a popup with the definition is shown. Figure 7.6 shows a high-level overview of the components in the NoteAid system, with example text that has been annotated. Users enter their EHR note text into NoteAid

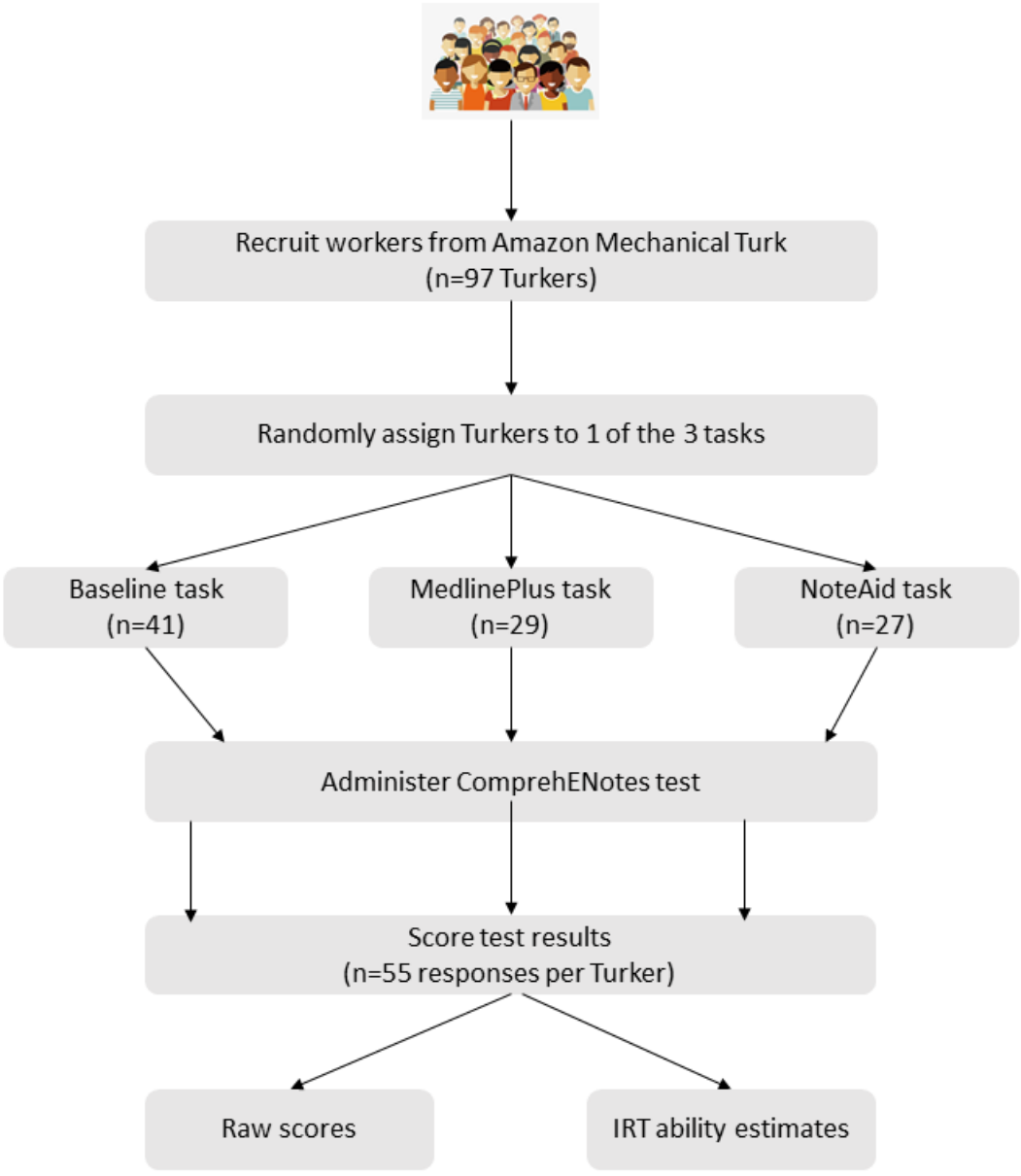


Figure 7.5: Flowchart describing our experiment. Amazon Mechanical Turk workers were randomly assigned to one of three tasks on the platform. They completed the ComprehENotes test with the use of the provided external tool. All scores were then collected, and ability estimated were obtained using Item Response Theory (IRT).

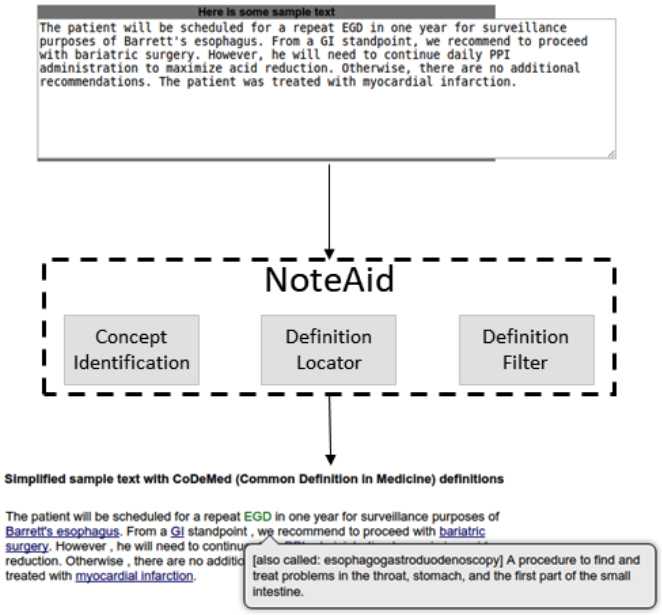


Figure 7.6: Example showing NoteAid simplified text.

and are provided with a reproduction of the text, with key medical concepts linked to their definitions.

NoteAid consists of 2 components. The concept identifier component processes input text and maps terms to medical concepts. The concepts are mapped to entries in the Unified Medical Language System using MetaMap [Aronson, 2001, Bodenreider, 2004]. It then filters the list of returned concepts to include only concepts that match a subset of possible semantic types related to patient health (e.g., disease or syndrome and lab or test result). The definition fetcher component uses the filtered list of concepts to pull definitions from an external knowledge resource (e.g., Wikipedia or MLP).

Previous evaluation of NoteAid has shown that patients' self-reported comprehension scores improve when using the system [Polepalli et al., 2013]. However, there has not yet been an evaluation of NoteAid on a test of comprehension, as opposed to self-reporting scores.

7.3.3 Item Response Theory Analysis

Recall that the ComprehENotes test set was developed using IRT [Baker and Kim, 2004]. The test set was built according to a single factor, 3-parameter logistic IRT model with a fixed guessing parameter. The test, therefore, measures a single latent trait, specifically the ability to comprehend EHR notes. Once the model has been fit, ability for a new test respondent is estimated by estimating θ according to the respondent's answers to the test questions after the responses have been converted to a correct or incorrect binary format. For a single test question i, the probability that individual j answers the question correctly is a function of the individual's ability (θ). The likelihood of a data set of response patterns is defined as:

$$p_i(\theta_j) = c_i + \frac{1 - c_i}{1 + e^{-a_i(\theta_j - b_i)}} \tag{7.1}$$

$$q_i(\theta_j) = 1 - p_i(\theta_j) \tag{7.2}$$

$$p(U_j|\theta_j) = \prod_{i=1}^{I} p_i(\theta_j)^{u_{ij}} q_i(\theta_j)^{(1-u_{ij})} \tag{7.3}$$

where Equation 7.1 is used to calculate the probability that individual j with an estimated ability of θ_j will answer question i correctly; Equation 7.2 calculates the probability that individual j with estimated ability θ_j will answer question i incorrectly; and Equation 7.3 calculates the likelihood of individual j's set of responses U_j to all items in the test set, where u_{ij} is 1 if individual j answered item i correctly and 0 if they did not.

p_i and q_i are functions of the known item parameters, and therefore, we can estimate θ via maximum likelihood for each Turker. We also calculated raw test scores for each Turker (percent of questions answered correctly) for comparison.

7.3.4 Results

We first report the demographic information for the Turkers who completed our tasks. Table 7.6 shows the demographic information that we collected from the Turkers for the Baseline, Treatment-MLP, and Treatment-NA groups. Overall, most of the Turkers who completed our tasks are white, young, and have at least an associate degree. In addition, most of the Turkers do not work in the medical field. These demographics are not representative of a wider population and do not fit demographics that are more commonly associated with low health literacy [Lynn Nielsen-Bohlman, 2004]. However, our goal here is to compare the results with respect to different interventions. In this case, we do not need to test individuals with low health literacy; we instead want to see if scores improve when users are provided with certain external resources.

Our analysis includes both the raw test scores as well as the estimated ability level using IRT. As the test set consists of questions that were fit using IRT, we can also calculate the ability of these Turkers and test whether the mean ability score was higher for Turkers that used NoteAid. Ability is a useful metric as it takes into consideration which questions you answer correctly, not just how many. IRT models question difficulty, so by considering whether easy or difficult answers were correct, IRT allows for a more informative score than percent correct. For each Turker, we calculated their ability score (θ) using the IRT model fit as part of the ComprehENotes data set. We use the mirt and ltm open-source R packages for estimation [Rizopoulos, 2006, Chalmers et al., 2012].

Figure 7.7 plots the raw scores for each AMT Turker for our test set. The center rectangles span the range from the first quartile to the third quartile of responses, and the bolded line inside each box represents the median score. Open circles indicate outlier scores. The upper horizontal line marks the maximum score for each group, and the lower horizontal line is 1.5 times the interquartile range below the first quartile.

Demographic	**Baseline** n (%), (n=41)	**MLP** n (%), (n=29)	**NA** n (%), (n=27)	**Total** n (%), (n=97)
Gender				
Male	27 (66)	8 (28)	18 (67)	53 (55)
Female	14 (34)	21 (72)	9 (33)	44 (45)
Age				
22-34	23 (56)	16 (55)	16 (59)	55 (57)
35-44	6 (15)	9 (31)	8 (30)	23 (24)
45-54	8 (20)	2 (7)	3 (11)	13 (13)
55-64	4 (10)	2 (7)	0 (0)	6 (6)
65 and older	0 (0)	0 (0)	0 (0)	0 (0)
Ethnicity				
Black or African American	8 (20)	3 (10)	4 (15)	15 (16)
Asian	3 (7)	0 (0)	1 (4)	4 (4)
Hispanic	4 (10)	1 (3)	0 (0)	5 (5)
American Indian or Alaska Native	0 (0)	1 (3)	1 (4)	2 (2)
White	26 (63)	24 (83)	21 (78)	71 (73)
Education				
Less than High School	1 (2)	0 (0)	0 (0)	1 (1)
High School Degree	9 (22)	8 (28)	8 (30)	25 (26)
Associate's Degree	8 (20)	5 (17)	3 (11)	16 (17)
Bachelor's Degree	20 (49)	14 (48)	14 (51)	48 (50)
Master's Degree or Higher	3 (7)	2 (7)	2 (7)	7 (7)
Occupation				
Physician	0 (0)	0 (0)	1 (4)	1 (1)
Nurse	2 (5)	0 (0)	0 (0)	2 (2)
Medical student	1 (2)	1 (3)	1 (4)	3 (3)
Other profession in medicine	2 (5)	3 (10)	3 (11)	8 (8)
Other profession	36 (88)	25 (86)	22 (82)	83 (86)

Table 7.6: Demographic information of Turkers from the follow-up study.

As the figure shows, visually there is a spread between the populations that did and did not have access to the interventions. Median raw scores for the baseline and MLP

groups are similar, whereas median scores for the NoteAid group is higher. The spread of responses for the treatment groups is also smaller than the baseline group.

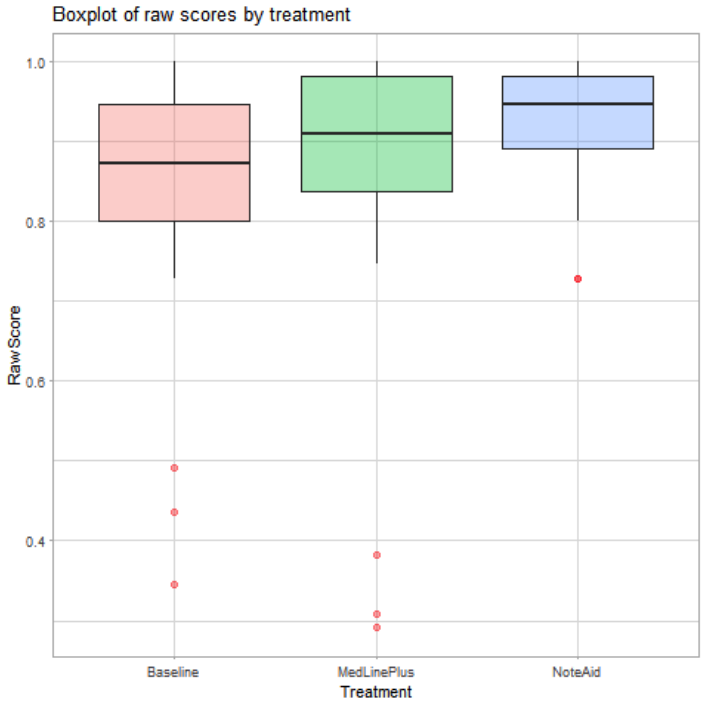


Figure 7.7: Box plot of raw scores for baseline and treatment Turker groups. The treatment groups were able to use MedlinePlus and NoteAid, respectively, when taking the ComprehENotes test.

Figure 7.8 shows the box plots of ability estimates. Again, the median values for the baseline and MLP groups are similar and the median ability estimates for the NoteAid group is higher. The lowest ability estimates for the baseline and MLP groups are much lower than for the NoteAid group (2 SDs below the mean as opposed to 1 SD below). This shows that even for individuals that use NoteAid and still struggle, the low range of ability is higher than when NoteAid is not used.

To test whether either intervention caused a significant difference in scores, we compared each intervention with our baseline using Welch 2-sample t test. Table 7.7 shows the mean raw scores and mean ability estimates for Turkers in each group.

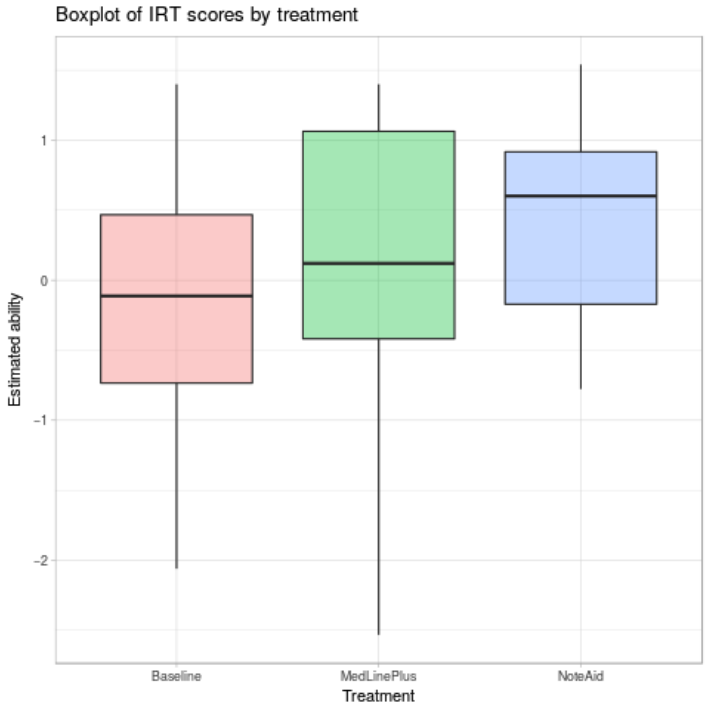


Figure 7.8: Box plot of ability estimates for baseline and treatment Turker groups. The treatment groups MLP and NA were able to use MedlinePlus and NoteAid, respectively, when taking the ComprehENotes test. IRT: Item Response Theory.

Mean scores are significantly higher than the baseline for Turkers that had access to NoteAid, both with regard to the raw scores (P=.01) and estimated ability (P=.02).

We also wanted to determine if demographic factors had an impact on test scores. To that end, we fit a linear regression model to predict raw scores using demographic information and group (e.g., baseline or treatment) as features. The results of the analysis showed that the intervention (none, MLP, or NoteAid) was a significant feature in predicting raw score. In addition, certain demographic groups were significant in determining score. Regarding ethnicity, individuals who self-reported as white had a significant positive coefficient. Regarding education, individuals that have less than a high school degree had a significant negative coefficient. These results are consistent with what is known about populations that are at risk for low health literacy. Individuals with lower education often have higher instances of low health

literacy, as well as minorities. Our populations for this task, particularly with regard to minorities and less educated individuals, were very small. Future work on NoteAid in minority populations would be worthwhile to confirm these effects.

Group	Raw score	Ability estimate
Baseline	0.831	-0.065
MedlinePlus	0.849	0.138
NoteAid	0.923*	0.477*

Table 7.7: Mean scores for the 3 groups. Mean NoteAid scores are significantly higher than the mean baseline scores, both for raw scores ($P = .01$) and estimated ability ($P = .02$).

7.3.5 Comparison With the S-TOFHLA

All Turkers who completed our tasks were also given the S-TOFHLA test to complete. Scores on S-TOFHLA place test-takers into 1 of the 3 categories: inadequate health literacy, marginal health literacy, and adequate health literacy. It is most useful as a screening tool to identify individuals with low or marginal health literacy. All Turkers in our tasks were scored to have adequate health literacy. In fact, all Turkers either scored perfect scores or only answered 1 question incorrectly, whereas the scores from the ComprehENotes test covered a wide range of ability estimates. The ComprehENotes can be used to assess EHR note comprehension at a more granular level as opposed to a screening tool such as S-TOFHLA, where the primary concern is identification of individuals with low health literacy.

7.3.6 ComprehENotes Analysis

Finally, we wanted to see if the IRT model that was originally fit as part of the ComprehENotes data set was validated by the response patterns that we collected from the Turkers. To this end, we selected the 2 questions that the most Turkers answered correctly as well as the 2 questions that the fewest Turkers answered correctly.

These questions can be considered the easiest and hardest, respectively, from our task. The difficulty parameters for these items as modeled by IRT match the expectation of how difficult these items should be. The 2 hardest questions from our task (in terms of how many Turkers answered correctly) have difficulty parameters of 0.7 and -0.3, whereas the 2 easiest questions have difficulty parameters of -1.8 and -1.4. The difficulty parameter is associated with the level of ability at which an individual has a 50% chance of answering the question correctly. Therefore, the low difficulty levels imply that someone of low ability has a 50% chance of answering the question correctly. Conversely, a higher difficulty parameter means that someone must be of a higher estimated ability level to have a 50% chance of answering correctly.

7.3.6.1 Discussion

We have shown the importance of targeted, active intervention when trying to improve a person's ability to comprehend EHR notes. By giving Turkers access to NoteAid, scores on the ComprehENotes test are significantly improved over a baseline population that had no external resources. On the other hand, Turkers that had access to MLP but had to search themselves for the information that they wanted did not have a significant improvement in scores. NoteAid automatically identifies key medical concepts and provides definitions, as opposed to the scenario with MLP, where a user must decide what to search for. The user may not know that a certain concept is key for understanding a passage or they may assume that they understand certain concepts that they do not. By letting the user decide what to search for, important terms may be missed and overall comprehension may be affected. This result is consistent with previous work on assessing comprehension using tools such as NoteAid [Polepalli et al., 2013], but this is the first time where the conclusion is based on an EHR note comprehension assessment instead of patient self-reported scores. By

using the ComprehENotes test, we can quantitatively confirm the previous results self-reported by patients.

There are limitations to this work. First, by using AMT, we are not able to monitor the Turkers who complete our task to ensure that only the external resources that we provide were used. This is particularly true in the baseline group, where our expectation is that no external resource was used. However, it is unlikely that the baseline users were able to access NoteAid without prior knowledge of the system; therefore, we can be confident that they did not use it in our task. If the baseline users did use external resources, they most likely used a passive resource such as Google or even MLP. As NoteAid was integrated into the Treatment-NA task, we can be confident that Turkers in the Treatment-NA task used NoteAid. The discrepancy between Treatment-MLP and Treatment-NA may seem to bias improvements toward the Treatment-NA group, but there is an important distinction to be made. At present, sites such as MLP are available to any patient that seeks them out, but the onus is on the patient to go to the site and search for terms. With the Treatment-NA group, we have shown that by integrating a system that can simplify and define medical terms automatically, the burden of defining terms is removed from the patient.

In addition, the demographics of the Turkers who completed our task are not representative of the larger population, specifically among demographics associated with higher risks of low health literacy [Lynn Nielsen-Bohlman, 2004]. In the case of this work, that is not problematic, as our goal was to examine the effect of active and passive interventions on EHR note comprehension. The demographics of our 3 groups were similarly distributed, so the changes in scores can be linked to the intervention used. Although the results obtained were significant, ideally larger populations could be examined in each group. However, as the demographics of the Turkers are not consistent with demographic groups associated with low health literacy, the follow-up work should focus on those groups. By using AMT and Turkers, we have shown that

tools such as NoteAid do improve EHR note comprehension generally, but future work should look specifically at groups associated with low health literacy to determine if our results hold for those groups as well.

Another limitation of this study is that patients are not evaluated on their own notes. Ideally, we would be able to assess the EHR note comprehension of each patient by testing the patient using concepts extracted from his or her own EHR notes. However, there are several roadblocks to making this a reality. First, this type of personalized assessment would reduce the ability to compare comprehension ability between patients. If a patient scores highly on an assessment of their own note, we can say that the patient understands the note, but if there were no complex concepts in the note, we cannot compare this with a patient who scores poorly on an evaluation based on his or her own complex EHR note. Second, to build a personalized EHR note evaluation would require complex natural language processing (NLP) systems to automatically generate multiple- choice questions (MCQs) for patients when they enter their EHR notes. To our knowledge, there does not currently exist an NLP system for medical MCQ generation. We do believe that the development of such a system will be beneficial for personalized patient assessment of EHR note comprehension. Such a personalized system could complement the ComprehENotes test so that a patient would be assessed on their own EHR note as well as on a standardized assessment.

We have shown that simply having access to resources designed to improve health literacy and medical concept understanding is not enough to provide benefit. The Turkers in our experiment who had access to MLP did not score significantly higher on the ComprehENotes test than those Turkers that were not provided with an external resource. On the other hand, having access to NoteAid, which actively pulls definition information and provides it to the user, led to significantly higher scores for Turkers. This result validates previously reported self-scored comprehension results showing

that users had an easier time understanding their notes when they had access to NoteAid.

Knowing that users do not see benefits from simply having access to MLP is an important observation. When doctors are recommending next steps for patients who wish to improve their health literacy, it may not be sufficient to point them to Web-based resources. Targeted interventions are necessary to ensure that patients are able to learn about specific concepts and diseases that are relevant to them. In particular, the integration of NoteAid with the EHR note on a patient's portal would remove the friction from the patient accessing an external resource. Instead, the patient would have key terms defined and simplified within his or her own patient portal, which would minimize the effort involved from the patient's standpoint and keep the information in the note within the portal itself.

There are several directions for future work. Developing target curricula is necessary to ensure that patients can see benefits from Web-based resources. They may not need a tool such as NoteAid (e.g., if they are not looking at their notes), but something more targeted than MLP is needed to ensure that patients are learning. In addition, there should be further validation of the ComprehENotes test set with patients that are at risk for low health literacy. The Turkers in our task all scored either close to average or above average in our ability estimates, except for a few outliers. The test was designed to be most informative for individuals of lower ability, so this test should be replicated with such a population.

CHAPTER 8

CONCLUSIONS

8.1 Contributions

In this dissertation we have made several contributions based on an analysis of the thesis statement:

Estimating the characteristics of individual data points such as difficulty and latent model ability using psychometric methods can be done at a large scale, can improve model performance, and can allow for more thorough model evaluation.

First, we have developed three new test sets using methods from IRT: two tests for the NLP tasks of natural language inference and sentiment analysis (Chapter 2), and a new test of electronic health record note comprehension (Chapter 7). Each test allows for a new way to analyze machine learning models (or patients in the case of the EHR note comprehension test) to better understand performance beyond a raw accuracy score. For NLP, we have shown that analyzing model performance using IRT can better measure performance on very easy/very hard data sets by comparing the model's performance to a population of humans. If a test set is easy, then very high accuracy is not as impressive, and the latent ability score is reflective of that. On the other hand, if a data set is very hard, even average performance in terms of accuracy can be indicative of a high-ability model. For EHR note comprehension, the ComprehENotes test is a measurement instrument that can assess patient EHR note comprehension using questions that come from real-world de-identified patient notes. The test questions cover several common diseases and conditions, and the questions in the test were analyzed and confirmed to be appropriate using IRT. What's more, using

the ComprehENotes test we were able to confirm previously self-reported results on patient comprehension improvement using the NoteAid tool with a valid measurement. By having individuals take the ComprehENotes test with and without NoteAid we can measure the impact of the NoteAid tool on EHR note comprehension instead of relying on self-reported patient scores.

Knowing how easy or difficult specific examples in a data set are is useful information for analyzing machine learning model performance. In particular, we analyzed the effect of (a) example difficulty and (b) model training set size to predict whether a trained model would label an example correctly and found that both difficulty and training set size are significant predictors of performance (Chapter 3). What's more, as more training data is added to a model, the odds of labeling an easy example correctly increase more quickly than the odds of labeling a difficult example correctly. This result shows that knowing the difficulty of examples in the data set allows for predictions in terms of model performance.

Next, we introduced the Soft Label Memorization Generalization (SLMG) algorithm to leverage disagreements between annotators to improve machine learning model performance (Chapter 4). Even though a relatively small number of soft-labeled examples were used, by incorporating the distribution over labels into training we were able to see improved model performance. For both natural language inference and sentiment analysis, using the soft-labeled data led to improvements in test set accuracy, indicating that the typical supervised learning paradigm of binary labeling is excluding useful information for learning.

A bottleneck of the results detailed above is that utilizing IRT requires human response pattern data. In Chapter 5 we show that human annotators can be replaced by an artificial crowd, allowing us to generate response pattern data for entire machine learning data sets instead of a carefully selected sample of examples. Using variational inference methods we are able to fit IRT models for all of the examples in the data

sets, and use the learned difficulty parameters to select the best examples in terms of difficulty for model training.

Finally, we introduced the Dynamic Data selection for Curriculum Learning via Ability Estimation (DDaCLAE) algorithm for dynamic curriculum learning (Chapter 6). By modeling model ability at each training epoch, we are able to train neural network models that are more efficient and more effective than baseline models and models trained with other curriculum learning strategies. DDaCLAE is the first curriculum learning strategy to measure model competency during training and use this information to inform data selection. Because DDaCLAE uses IRT, estimated competency can be compared directly with example difficulty to select training examples that the model has the highest likelihood of labeling correctly. DDaCLAE is dynamic, and can adjust whether more or less training data is included at a given epoch based on estimated ability at a specific point in the training process. This is in contrast to previous approaches to curriculum learning where the data inclusion rates are fixed and monotonically increasing.

We have have shown that the incorporation of Item Response Theory methods can benefit machine learning model training and evaluation. By using IRT to measure latent parameters of machine learning models and the examples that they are trained/tested on, researchers can better understand model performance on specific tasks, and the inherent difficulty of the tasks themselves without relying in heuristics. The work in this thesis should stimulate future work in supervised learning by encouraging researchers to fit IRT models of their data using an artificial crowd as part of the standard analysis. Learning how easy or difficult your data set is should become standard practice so that the relative importance of the task can be known by the community.

8.2 Future Work

As a result of this dissertation there are a number of interesting areas for future work:

8.2.1 Amortized IRT

We have shown that it is possible to learn IRT models with artificial crowds and variational inference methods. An interesting avenue for future work is to investigate whether example difficulty can be estimated as a function of the example features themselves. Recent work in variational autoencoders has shown that it is possible to untangle latent codes when encoding examples. If difficulty is one such code, then it is possible to learn the difficulty of an example without requiring a huge number of responses. Once an amortized IRT model has been trained (with a large response pattern data set), then for future examples difficulty could be estimated as a function of the example itself.

8.2.2 Synthetic, Difficult Data Generation

If one is able to encode an example and untangle difficulty from the resultant latent code, then the next step would be to decode a latent code and difficulty value back to a data point. Given some latent code and a difficulty value, then the IRT decoder model should output an example that is easy or difficult depending on the input difficulty value. This would allow for synthetic data generation where the difficulty can be specified, opening up entire new areas of synthetic curriculum learning research.

8.2.3 Merging Supervised Learning and IRT

The most exciting area of future work, and the hope we have for supervised learning in general, is a new standard of training algorithms that incorporates IRT directly into the learning process. As a model is trained on data, at each epoch model outputs can be stored. The learner is then updated based not only on a typical supervised

loss function, but also with a variational objective for learning difficulty estimates for each of the training examples. That way the model learns an IRT function and a classification function jointly. At test time, the model can not only output a predicted class, but also a predicted difficulty, which allows the user to see how easy or difficult the model thinks a given test example is. This can improve model interpretability and potentially guard against adversarial attacks, as this model should determine that the adversarial examples are more difficult than typical in-distribution examples.

BIBLIOGRAPHY

[Akaike, 1974] Akaike, H. (1974). A new look at the statistical model identification. In *Selected Papers of Hirotugu Akaike*, pages 215–222. Springer.

[Amiri, 2019] Amiri, H. (2019). Neural Self-Training through Spaced Repetition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 21–31.

[Amiri et al., 2017] Amiri, H., Miller, T., and Savova, G. (2017). Repeat before forgetting: Spaced repetition for efficient and effective training of neural networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2401–2410.

[Amiri et al., 2018] Amiri, H., Miller, T., and Savova, G. (2018). Spotting spurious data with neural networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 2006–2016.

[An et al., 2003] An, J., Lee, S., and Lee, G. G. (2003). Automatic Acquisition of Named Entity Tagged Corpus from World Wide Web. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 2*, ACL '03, pages 165–168, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Aronson, 2001] Aronson, A. R. (2001). Effective mapping of biomedical text to the umls metathesaurus: the metamap program. *Proceedings. AMIA Symposium*, pages 17–21.

[Aronson and Lang, 2010] Aronson, A. R. and Lang, F.-M. (2010). An overview of MetaMap: historical perspective and recent advances. *Journal of the American Medical Informatics Association*, 17(3):229–236. 00416.

[Aroyo and Welty, 2015] Aroyo, L. and Welty, C. (2015). Truth is a lie: Crowd truth and the seven myths of human annotation. *AI Magazine*, 36(1):15–24.

[Arroyo et al., 2010] Arroyo, I., Mehranian, H., and Woolf, B. P. (2010). Effort-based tutoring: An empirical approach to intelligent tutoring. In *Educational Data Mining 2010*.

[Bachrach et al., 2012] Bachrach, Y., Graepel, T., Minka, T., and Guiver, J. (2012). How to grade a test without knowing the answers - A bayesian graphical model for adaptive crowdsourcing and aptitude testing. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress.

[Baker et al., 1999] Baker, D. W., Williams, M. V., Parker, R. M., Gazmararian, J. A., and Nurss, J. (1999). Development of a brief test to measure functional health literacy. *Patient Educ Couns*, 38(1):33–42. 00773.

[Baker, 2001] Baker, F. B. (2001). *The basics of item response theory*. ERIC.

[Baker and Kim, 2004] Baker, F. B. and Kim, S.-H. (2004). *Item Response Theory: Parameter Estimation Techniques, Second Edition*. CRC Press.

[Baldry et al., 1986] Baldry, M., Cheal, C., Fisher, B., Gillett, M., and Huet, V. (1986). Giving patients their own records in general practice: experience of patients and staff. *British medical journal (Clinical research ed.)*, 292:596–598.

[Beltagy et al., 2016] Beltagy, I., Roller, S., Cheng, P., Erk, K., and Mooney, R. J. (2016). Representing Meaning with a Combination of Logical and Distributional Models. *Computational Linguistics*, 42(4):763–808.

[Bengio, 2012] Bengio, Y. (2012). Deep learning of representations for unsupervised and transfer learning. *ICML Unsupervised and Transfer Learning*, 27:17–36.

[Bengio et al., 2011] Bengio, Y., Bastien, F., Bergeron, A., Boulanger-lew, N., Breuel, T., Chherawala, Y., Cisse, M., Côté, M., Erhan, D., Eustache, J., Glorot, X., Muller, X., Lebeuf, S. P., Pascanu, R., Rifai, S., Savard, F., and Sicard, G. (2011). Deep learners benefit more from out-of-distribution examples. In *AISTATS*, pages 164–172.

[Bengio et al., 2009] Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum learning. In *Proceedings of the 26th International Conference on Machine Learning*, pages 41–48. ACM.

[Bingham et al., 2019] Bingham, E., Chen, J. P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., Singh, R., Szerlip, P., Horsfall, P., and Goodman, N. D. (2019). Pyro: Deep universal probabilistic programming. *The Journal of Machine Learning Research*, 20(1):973–978.

[Bock and Aitkin, 1981] Bock, R. D. and Aitkin, M. (1981). Marginal maximum likelihood estimation of item parameters: Application of an em algorithm. *Psychometrika*, 46(4):443–459.

[Bodenreider, 2004] Bodenreider, O. (2004). The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic acids research*, 32(suppl 1):D267.

[Bowman et al., 2015] Bowman, S. R., Angeli, G., Potts, C., and Manning, D. C. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642. Association for Computational Linguistics.

[Bronson et al., 1986] Bronson, D. L., Costanza, M. C., and Tufo, H. M. (1986). Using medical records for older patient education in ambulatory practice. *Medical care*, 24:332–339.

[Browne, 2001] Browne, M. W. (2001). An overview of analytic rotation in exploratory factor analysis. *Multivariate Behavioral Research*, 36(1):111–150.

[Bruce and Wiebe, 1999] Bruce, R. F. and Wiebe, J. M. (1999). Recognizing subjectivity: a case study in manual tagging. *Natural Language Engineering*, 5(02):187–205.

[Burrington-Brown et al., 2005] Burrington-Brown, J., Fishel, J., Fox, L., Friedman, B., Giannangelo, K., Jacobs, E., Lang, D., Lemery, C., Malchetske, B., Morgan, J., Murphy, K., Okamoto, C., Peterson, R., Robin, D., Smith, C., Sweet, D., Thomas, M., Wolter, J., Zallar, B., and AHIMA e-HIM Personal Health Record Work Group (2005). Defining the personal health record. AHIMA releases definition, attributes of consumer health record. *J AHIMA*, 76(6):24–25.

[Carlson and von Davier, 2013] Carlson, J. E. and von Davier, M. (2013). Item response theory. *ETS Research Report Series*, 2013(2):i–69.

[Caruana, 1995] Caruana, R. (1995). Learning many related tasks at the same time with backpropagation. *Advances in Neural Information Processing Systems*, pages 657–664.

[Chalmers et al., 2012] Chalmers, R. P. et al. (2012). mirt: A multidimensional item response theory package for the r environment. *Journal of Statistical Software*, 48(6):1–29.

[Chang et al., 2017] Chang, H.-S., Learned-Miller, E., and McCallum, A. (2017). Active bias: Training a more accurate neural network by emphasizing high variance samples. In *Advances in Neural Information Processing Systems*.

[Chapman et al., 2003] Chapman, K., Abraham, C., Jenkins, V., and Fallowfield, L. (2003). Lay understanding of terms used in cancer consultations. *Psychooncology*, 12(6):557–566.

[Chen et al., 2017] Chen, Q., Zhu, X., Ling, Z., Wei, S., Jiang, H., and Inkpen, D. (2017). Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, Vancouver. ACL.

[Chen and Thissen, 1997] Chen, W.-H. and Thissen, D. (1997). Local Dependence Indexes for Item Pairs Using Item Response Theory. *Journal of Educational and Behavioral Statistics*, 22(3):265–289.

[Coberly et al., 2010] Coberly, E., Boren, S. A., Davis, J. W., McConnell, A. L., Chitima-Matsiga, R., Ge, B., Logan, R. A., Steinmann, W. C., and Hodge, R. H. (2010). Linking clinic patients to internet-based, condition-specific information prescriptions. *Journal of the Medical Library Association : JMLA*, 98:160–164.

[Collins et al., 1988] Collins, A., Brown, J. S., and Newman, S. E. (1988). Cognitive apprenticeship: Teaching the craft of reading, writing and mathematics. *Thinking: The Journal of Philosophy for Children*, 8(1):2–10.

[Dagan et al., 2006] Dagan, I., Glickman, O., and Magnini, B. (2006). The PASCAL Recognising Textual Entailment Challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190. Springer. DOI: 10.1007/11736790_9.

[Davis et al., 1993] Davis, T. C., Long, S. W., Jackson, R. H., Mayeaux, E. J., George, R. B., Murphy, P. W., and Crouch, M. A. (1993). Rapid estimate of adult literacy in medicine: a shortened screening instrument. *Fam Med*, 25(6):391–395.

[Dawid and Skene, 1979] Dawid, A. P. and Skene, A. M. (1979). Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics*, pages 20–28.

[Delbanco et al., 2012] Delbanco, T., Walker, J., Bell, S. K., Darer, J. D., Elmore, J. G., Farag, N., Feldman, H. J., Mejilla, R., Ngo, L., Ralston, J. D., Ross, S. E., Trivedi, N., Vodicka, E., and Leveille, S. G. (2012). Inviting Patients to Read Their Doctors' Notes: A Quasi-experimental Study and a Look Ahead. *Ann Intern Med*, 157(7):461–470.

[Demartini et al., 2012] Demartini, G., Difallah, D. E., and Cudré-Mauroux, P. (2012). ZenCrowd: Leveraging Probabilistic Reasoning and Crowdsourcing Techniques for Large-scale Entity Linking. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 469–478, New York, NY, USA. ACM.

[Devlin et al., 2019] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

[Diviani et al., 2017] Diviani, N., Dima, A. L., and Schulz, P. J. (2017). A Psychometric Analysis of the Italian Version of the eHealth Literacy Scale Using Item Response and Classical Test Theory Methods. *Journal of Medical Internet Research*, 19(4):e114.

[Draper et al., 1986] Draper, J., Field, S., Thomas, H., and Hare, M. J. (1986). Should women carry their antenatal records? *British medical journal (Clinical research ed.)*, 292:603.

[Freund and Schapire, 1997] Freund, Y. and Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *J. Comput. Syst. Sci.*, 55(1):119–139.

[Fries et al., 2005] Fries, J. F., Bruce, B., and Cella, D. (2005). The promise of PROMIS: using item response theory to improve assessment of patient-reported outcomes. *Clin. Exp. Rheumatol.*, 23(5 Suppl 39):S53–57.

[Gaines et al., 2011] Gaines, J. K., Levy, L. S., and Cogdill, K. W. (2011). Sharing medlineplus®/medline® for information literacy education (smile): a dental public health information project. *Medical reference services quarterly*, 30:357–364.

[Geng, 2016] Geng, X. (2016). Label distribution learning. *IEEE Transactions on Knowledge and Data Engineering*, 28(7):1734–1748.

[Golodetz et al., 1976] Golodetz, A., Ruess, J., and Milhous, R. L. (1976). The right to know: giving the patient his medical record. *Archives of physical medicine and rehabilitation*, 57:78–81.

[Good et al., 2015] Good, B. M., Nanis, M., Wu, C., and Su, A. I. (2015). Microtask crowdsourcing for disease mention annotation in PubMed abstracts. *Pac Symp Biocomput*, pages 282–293.

[Gottlieb et al., 2015] Gottlieb, A., Hoehndorf, R., Dumontier, M., and Altman, R. B. (2015). Ranking Adverse Drug Reactions With Crowdsourcing. *J Med Internet Res*, 17(3).

[Guo et al., 2013] Guo, W., Li, H., Ji, H., and Diab, M. (2013). Linking tweets to news: A framework to enrich short text data in social media. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 239–249. Association for Computational Linguistics.

[Gururangan et al., 2016] Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S. R., and Smith, N. A. (2016). Annotation artifacts in natural language inference datg. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 2, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

[Hacohen and Weinshall, 2019] Hacohen, G. and Weinshall, D. (2019). On The Power of Curriculum Learning in Training Deep Networks. In *ICML 2019*. arXiv: 1904.03626.

[Halbach et al., 2016] Halbach, S. M., Enders, A., Kowalski, C., Pförtner, T.-K., Pfaff, H., Wesselmann, S., and Ernstmann, N. (2016). Health literacy and fear of cancer progression in elderly women newly diagnosed with breast cancer–a longitudinal analysis. *Patient education and counseling*, 99(5):855–862.

[Halevy et al., 2009] Halevy, A., Norvig, P., and Pereira, F. (2009). The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12.

[Hertz et al., 1976] Hertz, C. G., Bernheim, J. W., and Perloff, T. N. (1976). Patient participation in the problem-oriented system: a health care plan. *Medical care*, 14:77–79.

[Hinton et al., 2015] Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

[Hoffman et al., 2013] Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347.

[Homer et al., 1999] Homer, C. S., Davis, G. K., and Everitt, L. S. (1999). The introduction of a woman-held record into a hospital antenatal clinic: the bring your own records study. *The Australian & New Zealand journal of obstetrics & gynaecology*, 39:54–57.

[Honeyman et al., 2005] Honeyman, A., Cox, B., and Fisher, B. (2005). Potential impacts of patient access to their electronic care records. *Inform Prim Care*, 13(1):55–60.

[Hovy et al., 2013] Hovy, D., Berg-Kirkpatrick, T., Vaswani, A., and Hovy, E. (2013). Learning whom to trust with mace. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1120–1130. Association for Computational Linguistics.

[Inel and Aroyo, 2017] Inel, O. and Aroyo, L. (2017). Harnessing diversity in crowds and machines for better ner performance. In *European Semantic Web Conference*, pages 289–304. Springer.

[Inel et al., 2014] Inel, O., Khamkham, K., Cristea, T., Dumitrache, A., Rutjes, A., van der Ploeg, J., Romaszko, L., Aroyo, L., and Sips, R.-J. (2014). Crowdtruth: Machine-human computation framework for harnessing disagreement in gathering annotated data. In *International Semantic Web Conference*, pages 486–504. Springer.

[Irizarry et al., 2015] Irizarry, T., DeVito Dabbs, A., and Curran, C. R. (2015). Patient Portals and Patient Engagement: A State of the Science Review. *J. Med. Internet Res.*, 17(6):e148.

[Jimenez et al., 2014] Jimenez, S., Duenas, G., Baquero, J., Gelbukh, A., Bátiz, A. J. D., and Mendizábal, A. (2014). UNAL-NLP: Combining soft cardinality features for semantic textual similarity, relatedness and entailment. *SemEval 2014*, page 732.

[Jones et al., 1992] Jones, R. B., McGhee, S. M., and McGhee, D. (1992). Patient online access to medical records in general practice. *Health Bull (Edinb)*, 50(2):143–150. 00027.

[Jordan et al., 1999] Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233.

[Kajino et al., 2012] Kajino, H., Tsuboi, Y., and Kashima, H. (2012). A convex formulation for learning from crowds. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

[Kardash et al., 1988] Kardash, C. A., Royer, J. M., and Greene, B. A. (1988). Effects of schemata on both encoding and retrieval of information from prose. *Journal of Educational Psychology*, 80(3):324–329.

[Kardash and Scholes, 1995] Kardash, C. M. and Scholes, R. J. (1995). Effects of Preexisting Beliefs and Repeated Readings on Belief Change, Comprehension, and Recall of Persuasive Text. *Contemporary Educational Psychology*, 20(2):201–221.

[Kaushik and Lipton, 2018] Kaushik, D. and Lipton, Z. C. (2018). How much reading does reading comprehension require? a critical investigation of popular benchmarks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2018.

[Kim, 2014] Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751. Association for Computational Linguistics.

[Kingma and Welling, 2014] Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In *International Conference on Learning Representations*.

[Krizhevsky and Hinton, 2009] Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images. Technical report, Citeseer.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

[Kumar et al., 2010] Kumar, M. P., Packer, B., and Koller, D. (2010). Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, pages 1189–1197.

[Kutner et al., 2006] Kutner, M., Greenburg, E., Jin, Y., and Paulson, C. (2006). The Health Literacy of America's Adults: Results from the 2003 National Assessment of Adult Literacy. Technical Report 2006-483, National Center for Education Statistics.

[Lai and Hockenmaier, 2014] Lai, A. and Hockenmaier, J. (2014). Illinois-LH: A Denotational and Distributional Approach to Semantics. *SemEval 2014*, page 329.

[Lake et al., 2015] Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. (2015). Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338.

[Lake et al., 2013] Lake, B. M., Salakhutdinov, R. R., and Tenenbaum, J. (2013). One-shot learning by inverting a compositional causal process. In *Advances in neural information processing systems*, pages 2526–2534.

[Landis and Koch, 1977] Landis, J. R. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.

[LeCun et al., 1995] LeCun, Y., Bengio, Y., et al. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995.

[LeCun et al., 2015] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436.

[LeCun et al., 1998] LeCun, Y., Cortes, C., and Burges, C. J. (1998). MNIST handwritten digit database.

[Lerner et al., 2000] Lerner, E. B., Jehle, D. V., Janicke, D. M., and Moscati, R. M. (2000). Medical communication: do our patients understand? *Am J Emerg Med*, 18(7):764–766.

[Levy et al., 2014] Levy, O., Dagan, I., and Goldberger, J. (2014). Focused entailment graphs for open IE propositions. *Proc. CoNLL*.

[Liu et al., 2019] Liu, X., He, P., Chen, W., and Gao, J. (2019). Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, Florence, Italy. Association for Computational Linguistics.

[Lober et al., 2006] Lober, W., Zierler, B., Herbaugh, A., Shinstrom, S., Stolyar, A., Kim, E., and Kim, Y. (2006). Barriers to the use of a Personal Health Record by an Elderly Population. *AMIA Annu Symp Proc*, 2006:514–518.

[Lynn Nielsen-Bohlman, 2004] Lynn Nielsen-Bohlman, Allison M. Panzer, D. A. K.-E. C. o. H. L. (2004). *Health Literacy: A Prescription to End Confusion*. The National Academies Press, Washington, D.C.

[MacCallum et al., 1996] MacCallum, R. C., Browne, M. W., and Sugawara, H. M. (1996). Power analysis and determination of sample size for covariance structure modeling. *Psychological methods*, 1(2):130.

[Marelli et al., 2014] Marelli, M., Menini, S., Baroni, M., Bentivogli, L., Bernardi, R., and Zamparelli, R. (2014). A sick cure for the evaluation of compositional distributional semantic models. In *LREC*, pages 216–223.

[Martınez-Plumed et al., 2016] Martınez-Plumed, F., Prudêncio, R. B., Martınez-Usó, A., and Hernández-Orallo, J. (2016). Making sense of item response theory in machine learning. In *Proceedings of 22nd European Conference on Artificial Intelligence (ECAI), Frontiers in Artificial Intelligence and Applications*, volume 285, pages 1140–1148.

[Masys et al., 2002] Masys, D., Baker, D., Butros, A., and Cowles, K. E. (2002). Giving patients access to their medical records via the internet: the pcasso experience. *Journal of the American Medical Informatics Association : JAMIA*, 9:181–191.

[Mazor et al., 2012a] Mazor, K. M., Roblin, D. W., Williams, A. E., Greene, S. M., Gaglio, B., Field, T. S., Costanza, M. E., Han, P. K. J., Saccoccio, L., Calvi, J., Cove, E., and Cowan, R. (2012a). Health literacy and cancer prevention: two new instruments to assess comprehension. *Patient Educ Couns*, 88(1):54–60.

[Mazor et al., 2012b] Mazor, K. M., Rogers, H. J., Williams, A. E., Roblin, D. W., Gaglio, B., Field, T. S., Greene, S. M., Han, P. K. J., and Costanza, M. E. (2012b). The Cancer Message Literacy Tests: psychometric analyses and validity studies. *Patient Educ Couns*, 89(1):69–75.

[McMullen et al., 2011] McMullen, K. D., McConnaughy, R. P., and Riley, R. A. (2011). Outreach to improve patient education at south carolina free medical clinics. *Journal of consumer health on the Internet*, 15:117–131.

[Miller et al., 2000] Miller, N., Lacroix, E.-M., and Backus, J. E. (2000). Medlineplus: building and maintaining the national library of medicine's consumer health web service. *Bulletin of the Medical Library Association*, 88(1):11.

[Mortensen et al., 2015] Mortensen, J. M., Minty, E. P., Januszyk, M., Sweeney, T. E., Rector, A. L., Noy, N. F., and Musen, M. A. (2015). Using the wisdom of the crowds to find critical errors in biomedical ontologies: a study of SNOMED CT. *J Am Med Inform Assoc*, 22(3):640–648.

[Munkhdalai and Yu, 2017] Munkhdalai, T. and Yu, H. (2017). Neural semantic encoders. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 397–407, Valencia, Spain. Association for Computational Linguistics.

[Munro et al., 2010] Munro, R., Bethard, S., Kuperman, V., Lai, V. T., Melnick, R., Potts, C., Schnoebelen, T., and Tily, H. (2010). Crowdsourcing and language studies: the new generation of linguistic data. In *Proceedings of the NAACL HLT 2010 workshop on creating speech and language data with Amazon's Mechanical Turk*, pages 122–130. Association for Computational Linguistics.

[Nair and Hinton, 2010] Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

[Natesan et al., 2016] Natesan, P., Nandakumar, R., Minka, T., and Rubright, J. D. (2016). Bayesian Prior Choice in IRT Estimation Using MCMC and Variational Bayes. *Frontiers in Psychology*, 7.

[Nazi et al., 2013] Nazi, K. M., Hogan, T. P., McInnes, D. K., Woods, S. S., and Graham, G. (2013). Evaluating Patient Access to Electronic Health Records. *Medical Care*, 51:S52–S56.

[Neubig et al., 2017] Neubig, G., Dyer, C., Goldberg, Y., Matthews, A., Ammar, W., Anastasopoulos, A., Ballesteros, M., Chiang, D., Clothiaux, D., Cohn, T., Duh, K., Faruqui, M., Gan, C., Garrette, D., Ji, Y., Kong, L., Kuncoro, A., Kumar, G., Malaviya, C., Michel, P., Oda, Y., Richardson, M., Saphra, N., Swayamdipta, S., and Yin, P. (2017). Dynet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980*.

[Nguyen et al., 2016] Nguyen, J., Moorhouse, M., Curbow, B., Christie, J., Walsh-Childers, K., and Islam, S. (2016). Construct Validity of the eHealth Literacy Scale (eHEALS) Among Two Adult Populations: A Rasch Analysis. *JMIR Public Health and Surveillance*, 2(1):e24.

[Noblin et al., 2012] Noblin, A. M., Wan, T. T. H., and Fottler, M. (2012). The Impact of Health Literacy on a Patient's Decision to Adopt a Personal Health Record. *Perspect Health Inf Manag*, 9(Fall).

[Orlando and Thissen, 2000] Orlando, M. and Thissen, D. (2000). Likelihood-Based Item-Fit Indices for Dichotomous Item Response Theory Models. *Applied Psychological Measurement*, 24(1):50–64.

[Osborn et al., 2007] Osborn, C. Y., Weiss, B. D., Davis, T. C., Skripkauskas, S., Rodrigue, C., Bass, P. F., and Wolf, M. S. (2007). Measuring adult literacy in health care: performance of the newest vital sign. *Am J Health Behav*, 31 Suppl 1:S36–46.

[Parker et al., 1995] Parker, R. M., Baker, D. W., Williams, M. V., and Nurss, J. R. (1995). The test of functional health literacy in adults: a new instrument for measuring patients' literacy skills. *J Gen Intern Med*, 10(10):537–541.

[Passonneau and Carpenter, 2014] Passonneau, R. J. and Carpenter, B. (2014). The benefits of a model of annotation. *Transactions of the Association of Computational Linguistics*, 2:311–326.

[Paszke et al., 2017] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch.

[Pennington et al., 2014] Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

[Platanios et al., 2019] Platanios, E. A., Stretcu, O., Neubig, G., Poczos, B., and Mitchell, T. (2019). 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, Volume 1 (Long and Short Papers)*, pages 1162–1172, Minneapolis, Minnesota. Association for Computational Linguistics.

[Polepalli et al., 2013] Polepalli, B. R., Houston, T., Brandt, C., Fang, H., and Yu, H. (2013). Improving patients' electronic health record comprehension with noteaid. *Studies in health technology and informatics*, 192:714–718.

[Poliak et al., 2018] Poliak, A., Naradowsky, J., Haldar, A., Rudinger, R., and Van Durme, B. (2018). Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191.

[Ranganath et al., 2014] Ranganath, R., Gerrish, S., and Blei, D. (2014). Black box variational inference. In *Artificial Intelligence and Statistics*, pages 814–822.

[Rasch, 1960] Rasch, G. (1960). *Studies in mathematical psychology: I. Probabilistic models for some intelligence and attainment tests.* Nielsen & Lydiche, Oxford, England.

[Reading et al., 2017] Reading, S. R., Go, A. S., Fang, M. C., Singer, D. E., Liu, I.-L. A., Black, M. H., Udaltsova, N., Reynolds, K., and Investigators, t. A. a. R. F. i. A. F.-C. R. N. A. (2017). Health Literacy and Awareness of Atrial Fibrillation. *Journal of the American Heart Association*, 6(4):e005128.

[Ritter et al., 2017] Ritter, S., Barrett, D. G., Santoro, A., and Botvinick, M. M. (2017). Cognitive psychology for deep neural networks: A shape bias case study. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2940–2949. JMLR. org.

[Rizopoulos, 2006] Rizopoulos, D. (2006). ltm: An R Package for Latent Variable Modeling and Item Response Analysis. *Journal of Statistical Software, Articles*, 17(5):1–25.

[Roller and Stevenson, 2015] Roller, R. and Stevenson, M. (2015). Held-out versus Gold Standard: Comparison of Evaluation Strategies for Distantly Supervised Relation Extraction from Medline abstracts. In *Sixth International Workshop on Health Text Mining and Information Analysis (LOUHI)*, page 97.

[Ross and Lin, 2003] Ross, S. E. and Lin, C.-T. (2003). The effects of promoting patient access to medical records: a review. *J Am Med Inform Assoc*, 10(2):129–138.

[Royer, 2004] Royer, J. M. (2004). Uses for the sentence verification technique for measuring language comprehension. *Progress in Education*.

[Royer and Carlo, 1991] Royer, J. M. and Carlo, M. S. (1991). Assessing the Language Acquisition Progress of Limited English Proficient Students: Problems and New Alternative. *Applied Measurement in Education*, 4(2):85–113.

[Royer et al., 1987] Royer, J. M., Greene, B. A., and Sinatra, G. M. (1987). The Sentence Verification Technique: A Practical Procedure for Testing Comprehension. *Journal of Reading*, 30(5):414–422.

[Royer et al., 1979] Royer, J. M., Hastings, C. N., and Hook, C. (1979). A sentence verification technique for measuring reading comprehension. *Journal of Literacy Research*, 11(4):355–363.

[Sabou et al., 2012] Sabou, M., Bontcheva, K., and Scharl, A. (2012). Crowdsourcing Research Opportunities: Lessons from Natural Language Processing. In *Proceedings of the 12th International Conference on Knowledge Management and Knowledge Technologies*, i-KNOW '12, pages 17:1–17:8, New York, NY, USA. ACM.

[Sakaguchi and Van Durme, 2018] Sakaguchi, K. and Van Durme, B. (2018). Efficient online scalar annotation with bounded support. In *Proceedings of the 56st Annual Meeting of the Association for Computational Linguistics*.

[Schapira et al., 2012] Schapira, M. M., Fletcher, K. E., Hayes, A., Eastwood, D., Patterson, L., Ertl, K., and Whittle, J. (2012). The development and validation of the hypertension evaluation of lifestyle and management knowledge scale. *J Clin Hypertens (Greenwich)*, 14(7):461–466.

[Schwarz et al., 1978] Schwarz, G. et al. (1978). Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464.

[Sheldon, 1982] Sheldon, M. G. (1982). Giving patients a copy of their computer medical record. *The Journal of the Royal College of General Practitioners*, 32:80–86. KIE BoB Subject Heading: patient access to records, Full author name: Sheldon, MG.

[Shrivastava et al., 2016] Shrivastava, A., Gupta, A., and Girshick, R. (2016). Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 761–769.

[Simonyan and Zisserman, 2015] Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*.

[Snow et al., 2008] Snow, R., O'Connor, B., Jurafsky, D., and Ng, A. (2008). Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 254–263. Association for Computational Linguistics.

[Socher et al., 2013] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, D. C., Ng, A., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642. Association for Computational Linguistics.

[Srivastava et al., 2014] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

[Strubell et al., 2019] Strubell, E., Ganesh, A., and McCallum, A. (2019). Energy and Policy Considerations for Deep Learning in NLP. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

[Sudore et al., 2006] Sudore, R. L., Yaffe, K., Satterfield, S., Harris, T. B., Mehta, K. M., Simonsick, E. M., Newman, A. B., Rosano, C., Rooks, R., Rubin, S. M., Ayonayon, H. N., and Schillinger, D. (2006). Limited literacy and mortality in the elderly: the health, aging, and body composition study. *J Gen Intern Med*, 21(8):806–812.

[Tenney et al., 2019a] Tenney, I., Das, D., and Pavlick, E. (2019a). Bert rediscovers the classical nlp pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

[Tenney et al., 2019b] Tenney, I., Xia, P., Chen, B., Wang, A., Poliak, A., McCoy, R. T., Kim, N., Van Durme, B., Bowman, S. R., Das, D., et al. (2019b). What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

[Teolis, 2010] Teolis, M. G. (2010). A medlineplus kiosk promoting health literacy. *Journal of consumer health on the Internet*, 14:126–137.

[Theano Development Team, 2016] Theano Development Team (2016). Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688.

[Tokui et al., 2015] Tokui, S., Oono, K., Hido, S., and Clayton, J. (2015). Chainer: a next-generation open source framework for deep learning. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*.

[Torralba et al., 2008] Torralba, A., Fergus, R., and Freeman, W. T. (2008). 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 30(11):1958–1970.

[Tsuchiya, 2018] Tsuchiya, M. (2018). Performance impact caused by hidden bias of training data for recognizing textual entailment. In chair), N. C. C., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Hasida, K., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S., and Tokunaga, T., editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

[Weinshall et al., 2018] Weinshall, D., Cohen, G., and Amir, D. (2018). Curriculum learning by transfer learning: Theory and experiments with deep networks. In *International Conference on Machine Learning*, pages 5235–5243.

[Weiss et al., 2005] Weiss, B. D., Mays, M. Z., Martz, W., Castro, K. M., DeWalt, D. A., Pignone, M. P., Mockbee, J., and Hale, F. A. (2005). Quick Assessment of Literacy in Primary Care: The Newest Vital Sign. *Ann Fam Med*, 3(6):514–522.

[Wiebe et al., 1999] Wiebe, J. M., Bruce, R. F., and O'Hara, T. P. (1999). Development and use of a gold-standard data set for subjectivity classifications. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 246–253. Association for Computational Linguistics.

[Williams et al., 2018] Williams, A., Nangia, N., and Bowman, S. R. (2018). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*.

[Young et al., 2014] Young, P., Lai, A., Hodosh, M., and Hockenmaier, J. (2014). From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2(0):67–78.

[Zhai et al., 2013] Zhai, H., Lingren, T., Deleger, L., Li, Q., Kaiser, M., Stoutenborough, L., and Solti, I. (2013). Web 2.0-based crowdsourcing for high-quality gold standard development in clinical natural language processing. *J. Med. Internet Res.*, 15(4):e73.

[Zheng and Yu, 2017] Zheng, J. and Yu, H. (2017). Readability Formulas and User Perceptions of Electronic Health Records Difficulty: A Corpus Study. *Journal of Medical Internet Research*, 19(3):e59.