# Learning Latent Parameters without Human Response Patterns: Item Response Theory with Artificial Crowds

**John P. Lalor**[1]*, **Hao Wu**[2], **Hong Yu**[1,3,4]
[1]College of Information and Computer Sciences, UMass Amherst
[2]Department of Psychology and Human Development, Vanderbilt University
[3]Department of Computer Science, UMass Lowell
[4]Center for Healthcare Organization & Implementation Research, Bedford VA
john.lalor@nd.edu, hao.wu.1@vanderbilt.edu,
hong.yu@umassmed.edu

## Abstract

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 item 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 item difficulty does not match difficulty as estimated from the machine RPs.

## 1 Introduction

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, (e.g. Lalor et al., 2016; 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 another work, the difficulty of specific items was found to be a significant predictor of whether a model would classify the item correctly (Lalor et al., 2018).

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 data points (called "items") 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 (Lalor et al., 2016, 2018).

While IRT models are designed to be learned with human RPs for at most 100 items, 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 work 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,a) and can reach or surpass human-level performance on classification tasks (Lake et al., 2015). In addition, generating IRT data with deep learning models

*Current affiliation: Mendoza College of Business, University of Notre Dame

*Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 4249–4259,
Hong Kong, China, November 3–7, 2019. ©2019 Association for Computational Linguistics

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 item 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 item parameters, namely training set filtering, and show that using difficulty to sample training data outperforms baseline methods; (4) we provide a qualitative analysis of items 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 items. 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]

## 2 Fitting Item Response Theory Models

### 2.1 Traditional Item Response Theory

Here we briefly describe IRT and the specific model under consideration, the Rasch model (also known as the one-parameter logistic or 1PL model) (Rasch, 1960).

We refer the reader to (Baker, 2001; Baker and Kim, 2004) for additional details on IRT, and to (Martınez-Plumed et al., 2016; Lalor et al., 2016, 2018) for more details on previous applications of IRT to machine learning.

IRT models are designed to estimate latent ability parameters ($\theta$) of subjects and latent item parameters such as difficulty of items ($b$). For a 1PL model, the probability that subject $j$ will answer item $i$ correctly is a function of the subject's latent ability $\theta_j$ and the item's latent difficulty $b_i$:

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

The probability that subject $j$ will answer item $i$ incorrectly is:

$$p(y_{ij} = 0|\theta_j, b_i) = 1 - p(y_{ij} = 1|\theta_j, b_i) \qquad (2)$$

The likelihood of a data set of RPs $Y$ from $J$ subjects to a set of $I$ items is:

$$p(Y|\theta, b) = \prod_{j=1}^{J} \prod_{i=1}^{I} p(Y_{ij} = y_{ij}|\theta_j, b_i) \qquad (3)$$

For the 1PL model, the difficulty parameter represents the ability level at which the probability of an individual answering an item correctly is 50%. This occurs when item difficulty is equal to subject ability ($\theta_j = b_i$ in Eq. 1).

The item 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 item parameters are learned, subjects' $\theta$ 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).

### 2.2 IRT with Variational 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 $\theta$ 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:

$$q(\theta, b) = \prod_{j=1}^{J} \pi_j^\theta(\theta_j) \prod_{i=1}^{I} \pi_i^b(b_i) \qquad (4)$$

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

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

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

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

## 3 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.

**SNLI** The SNLI data set (Bowman et al., 2015) is a popular data set for the natural language inference task. Briefly, each example in the data set consists of two sentences in English: the premise and the hypothesis, and a corresponding label. The correct label is "entailment" if the premise implies the hypothesis, "contradiction" if the premise implies that the hypothesis must be false, and "neutral" if the premise implies neither the hypothesis nor its negation. SNLI consists of 550k/10k/10k training/validation/testing examples examples.

**SSTB** The Stanford Sentiment Treebank (SSTB) (Socher et al., 2013) is a collection of English phrases extracted from movie reviews with fine-grained sentiment annotations (very negative, negative, neutral, positive, very positive). In this work we focus on binary sentiment classification, using the SST-2 split of the data set, where neutral examples have been removed. The data set consists of 67k/873/1.8k training/validation/testing examples.

**Human RP Data** The human RP data sets for SNLI and SSTB were previously collected from Amazon Mechanical Turk (AMT) workers (Lalor et al., 2016, 2018). For a randomly selected sample of items from SNLI and SSTB, new labels were gathered from 1000 AMT workers (Turkers). Each Turker labeled each item, so that for each item there were 1000 new labels. For each Turker, a RP was generated by grading the provided labels against the known gold-standard label.

**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.

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, $x^i_{\text{train}}$. 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):

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

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 to have an impact on performance in terms of item difficulty (Lalor et al., 2018). Further investigations of network modifications is left for future work.

## 4 Methods

We conduct the following experiments: (i) a comparison of IRT parameters learned from human and machine RP data, using existing IRT data sets (Lalor et al., 2016, 2018) 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.

### 4.1 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 (Lalor et al., 2016).

A traditional Rasch model was fit with both MML and VI. MML was implemented in the R package mirt (Chalmers et al., 2015) and VI in Pyro (Bingham et al., 2018), 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 item 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.

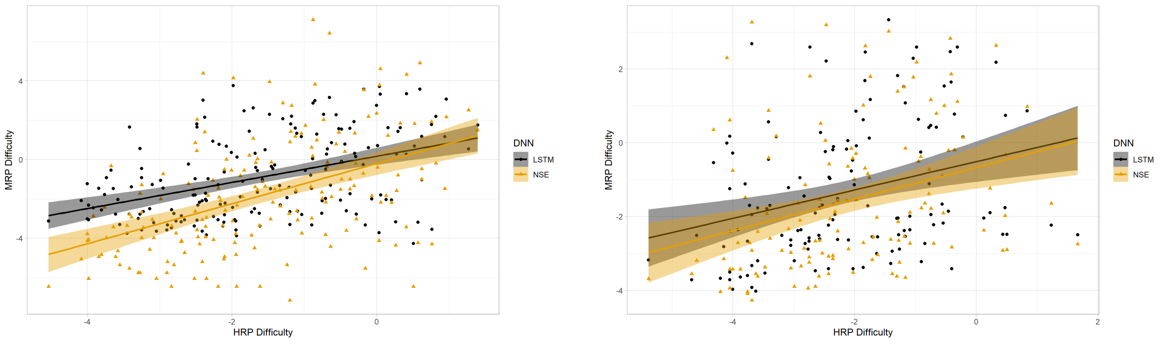
### 4.2 Human Machine Correlation

We further compare item 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 items that are easy for humans are also easy for machines. We fit two Rasch models, one with existing human RPs (Lalor et al., 2016, 2018). and one with the machine RPs. Both models were fit with MML using the mirt R package (Chalmers et al., 2015). Learned item difficulty parameters were extracted and compared via Spearman $\rho$ rank order correlations.

### 4.3 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 items 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 item difficulty threshold): (i) absolute value inner (AVI), where all training items with $|b_i| < d$ were retained, (ii) absolute value outer (AVO), where all training items with $|b_i| > d$



(a) NLI

(b) SA

Figure 1: Comparison of learned item difficulty parameters for human (x-axis) and machine data (y-axis) for NLI (Fig. 1a) and SA (Fig. 1b). Spearman $\rho$ (NLI): $0.409$ (LSTM) and $0.496$ (NSE). Spearman $\rho$ (SA): $0.332$ (LSTM) and $0.392$ (NSE).

were retained, (iii) an upper bound (UB), where items with $b_i < d$ were retained, and (iv) a lower bound (LB), where items with $b_i > d$ were retained. These methods were compared against two baselines that consider the percentage of models that label an item correctly ($0 \leq pc \leq 1$) as an inexpensive proxy for difficulty: (i) percent-correct upper bound (PCUB), where items with $pc_i < d$ were retained, and (ii) percent-correct lower bound (PCLB), where items 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 items 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 items from training.

Each of the filtering strategies have arguments in favor of their potential effectiveness. AVI includes "average" items in terms of training examples, none that are too easy or too difficulty. 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 Results

## 5.1 Human Machine Model Correlations

We first look at the results of our human-machine model comparison (Figures 1a and 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 $\rho$ values were 0.992 and 0.987 for SNLI and SSTB items, respectively.

For both SNLI and SSTB, we find a positive correlation between the item 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 DNN architectures has bearing on how the model identifies difficult items 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 items. 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., 2018) would further increase correlation scores.

## 5.2 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

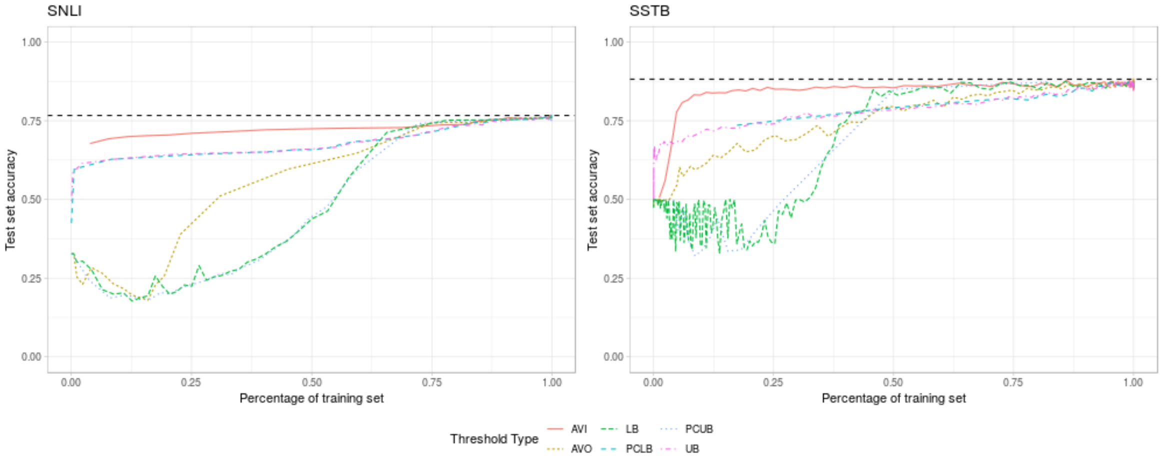


Figure 2: Test set accuracy by filtering strategy for NLI (left) and SA (right) plotted against percentage of training data retained. In both tasks filtering using the AVI strategy is most efficient in terms of high accuracy for small training set sizes.

parameters. Learned parameters are very similar between the two methods, which is to be expected. 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 (Lalor et al., 2016).

### 5.3 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 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 3). 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.

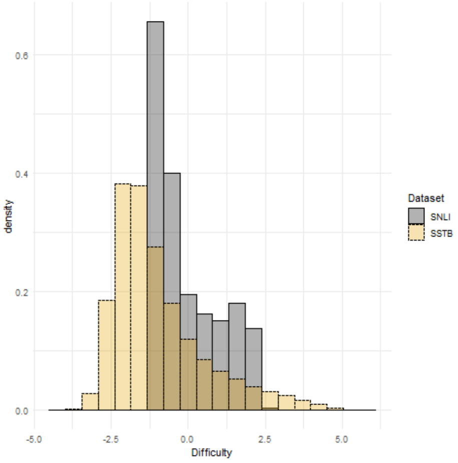


Figure 3: Density plot of learned difficulties for SNLI and SSTB data sets.

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 items were answered correctly/incorrectly) have an effect on item difficulty that is not captured by calculating percent correct.

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 item was included was the learned difficulty parameter $b_i$, which led to class imbalances in the training set. This imbalance, however did

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

| 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 1: Dev accuracy results for MT-DNN model with different training set sampling strategies.

not seem to have a significant negative effect in terms of performance. 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 random sample baseline, using the publicly available MT-DNN code.[3] Results are reported in Table 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.

## 6 Analysis

**Qualitative Evaluation of Difficulty** Table 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).

**Analysis of Differences** An interesting question comes up as a result of the less-than-perfect correlation scores (§5.1): 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 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 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 2, row 2) requires more abstract thinking than the earlier example. The humans are able to infer that because the girl is unwrapping an item, 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 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 item 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 item 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

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

| Premise | Hypothesis | Label | Difficulty |
|---|---|---|---|
| Two men and a woman are inspecting the front tire of a bicycle. | There are a group of people near a bike. | Entailment | -3.7 |
| 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. | Entailment | 3.1 |
| Two dogs playing in snow. | A cat sleeps on floor | Contradiction | -4.0 |
| Man sweeping trash outside a large statue. | A man is on vacation. | Contradiction | 3.8 |
| People sitting in chairs with a row flags hanging over them. | A family reunion for Fourth of July | Neutral | -3.6 |
| A group of dancers are performing. | The audience is silent. | Neutral | 3.8 |

Table 2: The easiest and hardest items judged by machine responses for each class in the SNLI test data set.

| Task | Label | Item Text | Difficulty ranking | | |
|---|---|---|---|---|---|
| | | | Humans | LSTM | NSE |
| SNLI | Contradiction | *P:* Two dogs playing in snow.<br>*H:* A cat sleeps on floor | 168 | 1 | 5 |
| | Entailment | *P:* A girl in a newspaper hat with a bow is unwrapping an item.<br>*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 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).

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.

## 7 Related Work

Prior work has considered IRT in the context of evaluating ML models using human (Lalor et al., 2016) and machine-generated (Martınez-Plumed et al., 2016) response patterns. Martınez-Plumed et al. (2016) attempted to fit IRT models using machine generated response patterns on small data sets (i.e. 200-300 items), but obtain results that are difficult to interpret using the existing IRT assumptions. Lalor et al. (2016) develop new IRT test sets for NLI using human-generated data, and present new ways to interpret and understand model performance beyond raw accuracy. Due to the need for human annotations the resulting tests are short (i.e. 124 examples). To the best of our knowledge no one has attempted to fit IRT models using DNN-generated response patterns on large data sets.

There have been a number of studies on modeling latent traits of data to identify a correct label, (e.g. Bruce and Wiebe, 1999). There has also been work in modeling individuals to identify poor annotators (Hovy et al., 2013), but neither jointly model the ability of individuals and data points, nor apply the resulting metrics to interpret DNN models. Other work has modeled the probability a label is correct along with the probability of an annotator to label an item correctly according to the (Dawid and Skene, 1979) model, but do not consider difficulty or discriminatory ability of the data points (Passonneau and Carpenter, 2014). In the above models an annotator's response depends on an item only through its correct label. IRT assumes a more sophisticated response mechanism involving both annotator qualities and item characteristics. The DARE model (Bachrach et al., 2012) jointly estimates ability, difficulty and response using probabilistic inference. It was evaluated on an intelligence test of 60 multiple choice questions administered to 120 individuals.

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 (Chang et al., 2017). In particular, the internal uncertainty of the model is used as the basis for selecting how training examples are weighted. However, model uncertainty depends upon the original training data the model was trained on, while here we use an external measure of uncertainty.

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 items 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).

## 8 Conclusion

In this work we have described how large-scale IRT models can be trained with DNN response patterns using VI. Learning the difficulty parameters of items 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 (additional results on two computer vision data sets are included in Appendix A).

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 items. 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 items 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 items 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. Prior work relied on human RPs to investigate the impact of difficulty on model performance (Lalor et al., 2018), but it is now possible to conduct similar IRT analyses with machine RPs. 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.

## Acknowledgements

We thank the anonymous reviewers for their comments and suggestions. This work was supported in part by the HSR&D award IIR 1I01HX001457 from the United States Department of Veterans Affairs (VA). We also acknowledge the support of LM012817 from the National Institutes of Health. This work was also supported in part by the Center for Intelligent Information Retrieval. The contents of this paper do not represent the views of CIIR, NIH, VA, or the United States Government.

## References

Hadi Amiri, Timothy Miller, and Guergana Savova. 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.

Yoram Bachrach, Thore Graepel, Tom Minka, and John Guiver. 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.

Frank B Baker. 2001. *The basics of item response theory*. ERIC.

Frank B. Baker and Seock-Ho Kim. 2004. *Item Response Theory: Parameter Estimation Techniques, Second Edition*. CRC Press.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th International Conference on Machine Learning*, pages 41–48. ACM.

E. Bingham, J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. Szerlip, P. Horsfall, and N. D. Goodman. 2018. Pyro: Deep Universal Probabilistic Programming. *ArXiv e-prints*.

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

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and D. Christopher Manning. 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.

Rebecca F. Bruce and Janyce M. Wiebe. 1999. Recognizing subjectivity: A case study in manual tagging. *Nat. Lang. Eng.*, 5(2):187–205.

Phil Chalmers, Joshua Pritikin, Alexander Robitzsch, and Mateusz Zoltak. 2015. mirt: Multidimensional Item Response Theory.

Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. 2017. Active bias: Training a more accurate neural network by emphasizing high variance samples. In *Advances in Neural Information Processing Systems*.

A. Philip Dawid and Allan M. Skene. 1979. Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):20–28.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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

Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. 2013. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 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.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. 1999. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233.

Divyansh Kaushik and Zachary C Lipton. 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.

Diederik P Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *International Conference on Learning Representations*.

M Pawan Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, pages 1189–1197.

Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. 2015. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338.

John P Lalor, Hao Wu, Tsendsuren Munkhdalai, and Hong Yu. 2018. Understanding deep learning performance through an examination of test set difficulty: A psychometric case study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2018.

John P. Lalor, Hao Wu, and Hong Yu. 2016. Building an evaluation scale using item response theory. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 648–657. Association for Computational Linguistics.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*.

Fernando Martınez-Plumed, Ricardo BC Prudêncio, Adolfo Martınez-Usó, and José Hernández-Orallo. 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.

Tsendsuren Munkhdalai and Hong Yu. 2017. Neural semantic encoders. *EACL 2017*.

Prathiba Natesan, Ratna Nandakumar, Tom Minka, and Jonathan D Rubright. 2016. Bayesian prior choice in irt estimation using mcmc and variational bayes. *Frontiers in psychology*, 7:1422.

Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, Kevin Duh, Manaal Faruqui, Cynthia Gan, Dan Garrette, Yangfeng Ji, Lingpeng Kong, Adhiguna Kuncoro, Gaurav Kumar, Chaitanya Malaviya, Paul Michel, Yusuke Oda, Matthew Richardson, Naomi Saphra, Swabha Swayamdipta, and Pengcheng Yin. 2017. Dynet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980*.

Rebecca J. Passonneau and Bob Carpenter. 2014. The benefits of a model of annotation. *Transactions of the Association of Computational Linguistics*, 2:311–326.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Rajesh Ranganath, Sean Gerrish, and David Blei. 2014. Black box variational inference. In *Artificial Intelligence and Statistics*, pages 814–822.

Georg Rasch. 1960. Studies in mathematical psychology: I. probabilistic models for some intelligence and attainment tests.

Keisuke Sakaguchi and Benjamin Van Durme. 2018. Efficient online scalar annotation with bounded support. In *Proceedings of the 56st Annual Meeting of the Association for Computational Linguistics*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, D. Christopher Manning, Andrew Ng, and Christopher Potts. 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.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. Bert rediscovers the classical nlp pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, et al. 2019b. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.