FHI Oxford Technical Report #2018-2

Predicting Human Deliberative Judgments with Machine Learning

Owain Evans[*], Andreas Stuhlmüller[†], Chris Cundy[*], Ryan Carey[†], Zachary Kenton[*], Thomas McGrath[*], Andrew Schreiber[†]

July 13, 2018

Abstract

Machine Learning (ML) has been successful in automating a range of cognitive tasks that humans solve *effortlessly* and *quickly*. Yet many real-world tasks are *difficult* and *slow*: people solve them by an extended process that involves analytical reasoning, gathering external information, and discussing with collaborators. Examples include medical advice, judging a criminal trial, and providing personalized recommendations for rich content such as books or academic papers.

There is great demand for automating tasks that require deliberative judgment. Current ML approaches can be unreliable: this is partly because such tasks are intrinsically difficult (even AI-complete) and partly because assembling datasets of deliberative judgments is expensive (each label might take hours of human work). We consider addressing this data problem by collecting *fast* judgments and using them to help predict deliberative (*slow*) judgments. Instead of having a human spend hours on a task, we might instead collect their judgment after 30 seconds or 10 minutes. These fast judgments are combined with a smaller quantity of slow judgments to provide training data. The resulting prediction problem is related to semi-supervised learning and collaborative filtering.

We designed two tasks for the purpose of testing ML algorithms on predicting human deliberative judgments. One task involves Fermi estimation (back-of-the-envelope estimation) and the other involves judging the veracity of political statements. We collected a dataset of 25,000 judgments from more than 800 people. We define an ML prediction task for predicting deliberative judgments given a training set that also contains fast judgments. We tested a variety of baseline algorithms on this task.

Unfortunately our dataset has serious limitations. Additional work is required to create a good testbed for predicting human deliberative judgments. This technical report explains the motivation for our project (which might be built on in future work) and explains how further work can avoid our mistakes. Our dataset and code is available at https://github.com/oughtinc/psj.

[*]University of Oxford
[†]Ought Inc.

1 Introduction

1.1 Fast and slow judgments

Machine Learning has been successful in automating mental tasks that are quick and effortless for humans. These include visual object recognition, speech recognition and production, and basic natural language prediction and comprehension [1, 2, 3, 4]. Andrew Ng states the following heuristic [5]:

> If a typical person can do a mental task with less than one second of thought, we can probably automate it using AI either now or in the near future.[1]

In this technical report we refer to judgments that are quick (roughly 30 seconds or less) and easy for most humans as **fast** judgments. Fast judgments contrast with **slow** judgments, which may involve lengthy processes of deliberate reasoning, research, experimentation, and discussion with other people.[2] Many important real-world tasks depend on slow judgments:

- Predict the verdict of jury members in a criminal trial.
- Predict which engineers will be hired by a company with an extensive interview process.
- Predict whether experts judge a news story to be fake or intentionally misleading.
- Predict a doctor's advice to an unwell patient after a thorough medical exam.
- Predict how a researcher will rate a new academic paper after reading it carefully.
- Predict how useful a particular video lecture will be for someone writing a thesis on recent Chinese history.

There is great demand for Machine Learning (ML) and other AI techniques for predicting human slow judgments like these, especially in hiring workers, detection of fake or malicious content, medical diagnosis, and recommendation [8, 9, 10, 11]. However ML approaches to predicting these slow judgments are often unreliable [12, 13, 14]: even if they do reasonably well on a majority of instances, they may have large errors on more demanding cases (e.g. on inputs that would be tricky for humans or on inputs chosen by humans to try to fool the algorithm).

One source of the unreliability for ML algorithms is optimizing for a subtly wrong objective. Suppose a student gets video lecture recommendations from

[1] Ng intends it as a heuristic rather than a rigorous scientific conclusion.

[2] The distinction is similar to that between System 1 (fast) and System 2 (slow) cognition [6, 7]. However, in this work we distinguish judgments by how long they take and whether they make use of external information and not by the underlying cognitive process. For instance, fast judgments can depend on quick application of analytical reasoning.

YouTube.[3] These recommendations may be optimized based on video popularity ("Do users click on the video?") and engagement ("Do users Like or share the video?"). These metrics are mostly based on *fast* judgments. Yet the student would prefer recommendations based on *slow* judgments, such as the evaluation of another student who has carefully checked the lecture for factual accuracy by tracking down the sources. Fast and slow judgments will sometimes coincide, as when a lecture is inaudible or off-topic. Yet a lecture may seem useful on first glance while turning out to be riddled with inaccuracies that careful research would expose.[4]

Predicting slow judgments in the tasks above is challenging for current ML algorithms. This is due in part to the intrinsic difficulty of the tasks; predicting how a student evaluates a lecture is arguably AI-complete [16]. Another difficulty is that collecting slow judgments is inherently expensive: if it takes five hours of fact-checking to recognize the errors in a lecture then a dataset of millions of such evaluations is impractically expensive. Big datasets won't solve AI-complete problems but will improve ML performance.

Predicting slow judgments is also related to long-term AI Safety, i.e. the problem of creating AI systems that remain aligned with human preferences even as their capabilities exceed those of humans [17, 18, 19, 20, 21]. Rather than create AI that shares only human goals, a promising alternative is to create AI that makes decisions in the way a human would at every timestep [22, 23]. This approach of imitating human decision-making is only promising if it imitates human deliberate (slow) judgments [24, 25]. A system making thousands of human-like slow judgments per second could have super-human capabilities while remaining interpretable to humans [26, 27].

1.2 Using Fast Judgments to Predict Slow Judgments

How can we get around the challenges of predicting slow judgments? One approach is to tackle the AI-completeness head on by trying to emulate the process of human deliberate reasoning.[5] A second approach (which complements the first) is to tackle the data problem head on and find ways to collect huge datasets of real or synthetic slow judgments [38]. This technical report explores an indirect approach to the data problem. Instead of collecting a big dataset of slow judgments, we collect a small dataset of slow judgments along with a larger quantity of *side information* related to the slow judgments.

What kind of side information would help predict a person's slow judgment? If Alice makes a slow judgment about a question, then Alice's *fast* judgment about the same question is relevant side information. As noted above, in predicting a student's thorough evaluation of a video lecture, it is helpful to know

[3] This is just meant as an example of a recommender system and is not a comment on the actual YouTube algorithm. A paper [15] on YouTube recommendations states that they optimize for whether users follow a recommendation (weighted by user watch-time). This will mostly depend on fast judgments.

[4] It could be argued that YouTube, Facebook and other sites are optimizing for being entertaining and keeping users on the site and that these are well predicted by fast judgments. Yet it's clear that users sometimes seek content that they would rate highly after a slow judgment (e.g. for educational purposes, for help making a business decision). So the question remains how to build ML algorithms for this task.

[5] There is a large and varied literature aiming to create algorithms that perform sophisticated reasoning. Here are some recent highlights: [28, 29, 30, 31, 32, 33, 34, 35, 36, 37]

their fast judgment (e.g. after watching the video for only 30 seconds). Likewise, a doctor's guess about a diagnosis after 30 seconds will sometimes predict their careful evaluation. Another kind of side information for predicting Alice's slow judgment about a question are the judgments of other people about the same question. This is the idea behind collaborative filtering [39], where a person's rating of a song is predicted from the ratings of similar people.

While collaborative filtering has been widely studied and deployed, there is little prior work on using fast judgments as side information for slow judgments. The motivation for using fast judgments is that they are often easily available and their cost is much lower than slow judgments. Human cognition is like an "anytime" iterative algorithm: whereas our slow judgments are more discerning and reliable, our fast judgments are well-typed and provide increasingly good approximations to slow judgments over time.[6] For most judgment tasks we can collect fast judgments from humans and these fast judgments will come at a cost orders of magnitude cheaper than slow judgments. Where fast judgments sometimes coincide with slow judgments and include uncertainty information[7], it may be better to spend a fixed budget on a mix of slow and fast judgments than on slow judgments alone. Often fast judgments are not just cheaper to collect but are essentially free. YouTube, Facebook, Twitter and Reddit have vast quantities of data about which content people choose to look at, share with others, "Like", or make a quick verbal comment on.

Using fast judgments as side information to predict slow judgments requires modifying standard ML algorithms. While the objective is predicting slow judgments, most of the training examples (e.g. most video lectures) only come with fast judgments. This is related to semi-supervised learning [40, 41], distant supervision [42] (where the training labels are a low-quality proxy for the true labels), learning from privileged information [43], as well as to collaborative filtering.

1.3 Contributions and caveats

This tech report describes a project applying machine learning (ML) to predicting slow judgments. We designed tasks where slow judgments (deliberate thinking and research) are required for doing well but quick judgments often provide informative guesses. We collected a dataset of fast and slow human judgments for these tasks, and formulated a set of prediction problems (predicting held-out slow judgments given access to varying quantities of slow judgments at training time). We applied ML baselines: standard collaborative filtering algorithms, neural collaborative filtering specialized to our tasks, and a Bayesian hierarchical model.[8].

Unfortunately our dataset turned out to be problematic and is unlikely to be a good testing ground for predicting slow judgments. This report describes parts of our project that may be usefully carried over to future work and summarizes what we learned from our efforts to create an appropriate dataset. The problem of predicting slow judgments (and of creating a dataset for the purpose)

[6] In doing a long calculation we might have no idea of the answer until we solve the whole problem. For many real-world problems our quick guesses are somewhat accurate and gradually improve with time.

[7] That is, the human provides both a guess and a measure of confidence.

[8] Code is available at https://github.com/oughtinc/psj

Fermi comparisons	Politifact truth judgments
weight of a blue whale in kg $<$ 50,000	*Rob Portman*: "Since the stimulus package was passed, Ohio has lost over 100,000 more jobs."
population of Moscow $*$ smaller angle in degrees between hands of clock at 1.45 $<$ 15,000,000	*Republican Party*: "Charlie Crist is embroiled in a fraud case for steering taxpayer money to a de facto Ponzi scheme."
driving distance in miles between London and Amsterdam $<$ 371	*Mark Zaccaria*: "James Langevin voted to spend $3 billion for a jet engine no one wants."
weight of $600 worth of quarters in pounds $*$ area of Canada in square miles $<$ 328,000,000	

Figure 1: Example questions. In Fermi, people guess whether the left-hand side is smaller than right-hand side. In Politifact they guess whether the speaker's statement is true.

was harder than expected. We hope to stimulate future work that avoids the problems we encountered. Our main contributions are the following:

1. We designed two slow judgment tasks for humans: **Fermi Comparisons** involve mental reasoning and calculation, and **Politifact Truth Judgments** involve doing online research to assess the veracity of a political statement.[9] We created a web app to collect both fast and slow judgments for the task. These tasks and the app could be used in future work.

2. We relate predicting slow judgments with side information to collaborative filtering and we implement simple baselines based on this relation.

3. We diagnose problems with our dataset. First, while slow judgments were significantly more accurate than fast judgments, the difference was smaller than expected. Second, variability among subjects was hard to distinguish from noise; so ML algorithms could not exploit similarities among users as in collaborative filtering. Third, while there is clear variation in how users respond to different questions, this variation is very hard for current ML algorithms to exploit.

2 Tasks and Datasets for Slow Judgments

Our overall goal was (A) to define the ML problem of predicting slow judgments given a training set of fast and slow judgments, and (B) to find or create datasets which can be used to test algorithms for this problem.

The domain that humans are making fast/slow judgments about should be AI-complete or closely related to an AI-complete problem.[10] Solving such a

[9] Both tasks require humans to decide some objective matter of fact. Yet there is no requirement that slow judgments be about objective facts: e.g. a student's judgment about a lecture is partly based on their own preferences and interests.

[10] The important real-world problems of predicting slow judgments in Section 1.1 are plausibly AI-complete.

task will (for some problem instances) depend on patterns or structures that current Machine Learning algorithms do not capture. So while we cannot hope for highly accurate predictions of slow judgments, we can seek ML algorithms that "know what they know" [44]. Such algorithms exploit patterns they are able to learn (producing confident accurate predictions) and otherwise provide well-calibrated uncertain predictions [45, 46].

This section describes the AI-complete tasks we designed for testing algorithms for predicting slow judgments. Before that we first review related datasets in this area.

2.1 Existing Datasets

Many ML datasets record human slow judgments for AI-complete tasks. These include datasets of movie reviews [47], reviewer scores for academic papers [48], and verdicts for legal cases [49]. There are also datasets where the ground-truth labels are the product of extensive cognitive work by humans (e.g. theorem proving [38], political fact-checking [14]) and these could potentially be used to study human slow judgments. However, these datasets do not contain fast judgments. In particular, for each task instance in the dataset, there's a slow judgment by a particular individual but there's no corresponding fast judgment. Moreover the datasets do not explicitly record information about the time and resources than went into the slow judgment[11]. For example, the reviewers of academic papers do not divulge whether they spent ten minutes or two hours reviewing a paper.

Due to the lack of existing datasets, we created a new dataset recording human slow and fast judgments for AI-complete tasks. We designed two tasks for this purpose, which may be useful for future work in this area (even if our dataset is not).

2.2 Two Judgment Tasks: Fermi and Politifact

We created two tasks for humans that require deliberate, multi-step thinking and rely on broad world knowledge. In the **Fermi Comparisons** task (abbreviated to "Fermi") users determine which of two quantities is larger. This often involves simple arithmetic, factual knowledge, and doing back-of-the-envelope estimates to determine the order of magnitude of the quantity. Example questions are show in Figure 1. Note that human subjects (who we refer to as "users") are not allowed to look up the quantities online as this would trivialize the task.

In the **Politifact Truth** task (abbreviated to "Politifact") users evaluate whether a statement by a US politician or journalist is true or false (see Figure 1). They have access to metadata about the statement (see Figure 2) and are allowed to do research online. For both tasks, users assign a probability that the statement is true (with "0" being definitely false and "1" being definitely true) and they enter their probability via the interface in Figure 2. Their goal is to minimize mean squared error between their probabilistic judgment and the ground-truth answer.

[11] A paper [50] on adversarial examples for humans does record human judgments under different time constraints. But this domain is not AI-complete and the slower judgments are still pretty fast.

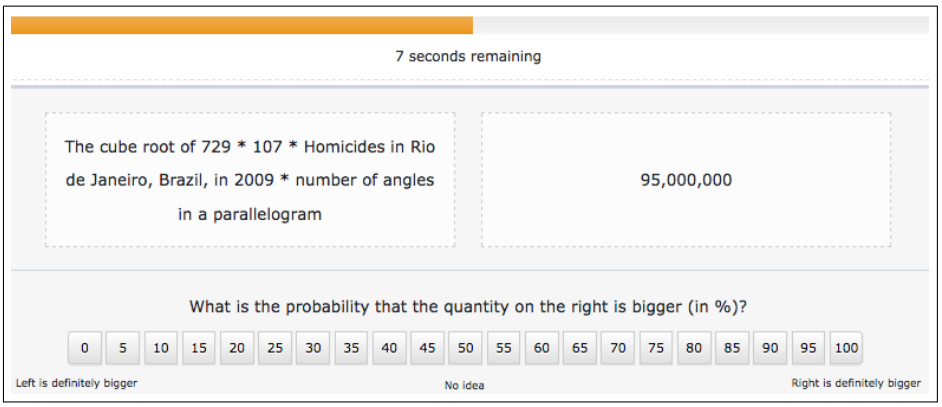
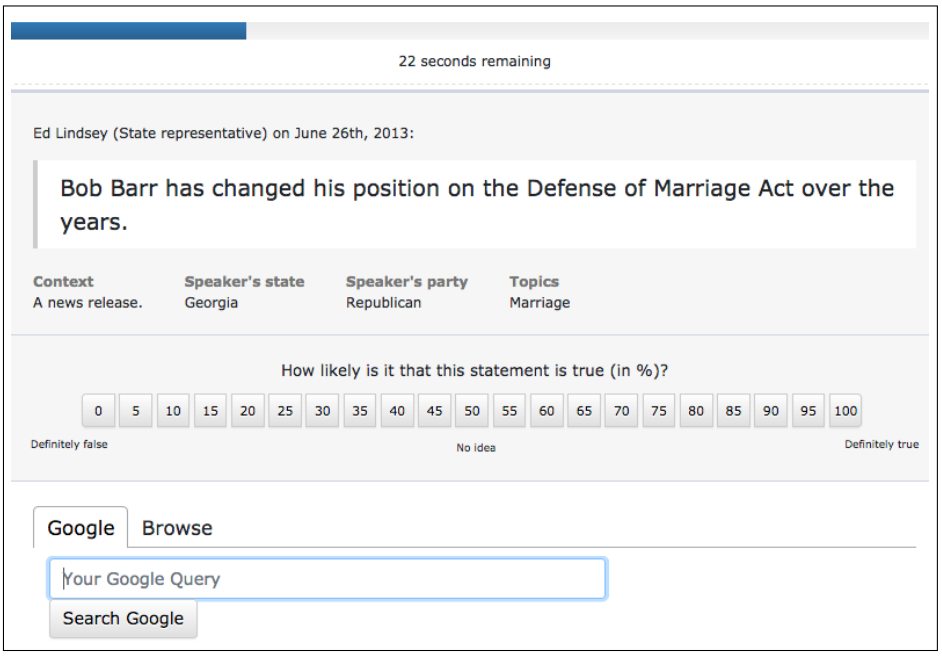


Figure 2: Question-answering UI for Fermi Comparisons (above) and Politifact Truth (below) tasks. Users are not allowed to do online research for Fermi but are for Politifact.



Questions and ground-truth answers for Fermi were constructed and computed by the authors. Political statements and ground-truth (i.e. the judgments of professional fact-checkers) for Politifact were downloaded from the politifact.com API (following [14]).

The Fermi and Politifact tasks satisfy the following desirable criteria:

- *A generalized version of each task is AI-complete.* Fermi questions depend on mathematical reasoning about sophisticated and diverse world knowledge (e.g. estimate the mass of the atmosphere). If novel Fermi questions are presented at test time (as in a job interview for high-flying undergraduates), the task is AI-complete. Politifact questions can be about any political topic (e.g. economics, global warming, international relations).

Answering them requires nuanced language understanding and doing sophisticated research on the web.

- *Fast judgments are informative about slow judgments.* Given a question (as in Figure 2), fast judgments of the probability (e.g. less than 30 seconds) will be informative about a user's slow judgment (e.g. 5 or 10 minutes). However some questions are too difficult to solve with a fast judgment.

- *It is possible to make progress through analytical thinking or gathering evidence.* For Fermi, breaking down the problem into pieces and coming up with estimates for each of the pieces is a useful cognitive strategy. For Politifact (but not Fermi), participants are allowed to use a search engine and to read web pages to check political facts.

- *The ground-truth is available for both Fermi and Politifact questions.* The problem of predicting slow judgments we discussed in the Introduction is not limited to human judgments about objective facts (as the list of examples in Section 1.1 makes clear). However, if the ground-truth is available then this makes certain experiments possible.[12]

2.3 Data collection

Human participants (who we refer to as "users") were recruited online[13] and answered questions using the UI in Figure 2. Users see a question, a form for entering their probability judgment, a progress bar that counts down the time remaining, and (for Politifact only) a search engine tool.

For each question, users provides fast, medium and slow probability judgments. In Fermi the user is presented with a question and has 15 seconds to make the *fast* judgment. Once the 15 seconds have elapsed, they get an additional 60 seconds in which they are free to change their answer as they see fit (the *medium* judgment). Finally they get another 180 seconds to make the *slow* judgment. The setup for Politifact is identical but the time periods are 30, 90 and 360 seconds. (Users are free to use less than the maximal amount of time. So for particularly easy or difficult questions, the "slow" judgment may actually only take 30 seconds or less.)

[12] In Politifact, the ground-truth is just an especially slow judgment of an expert in political fact-checking and could be modeled as such. We did not try this in our experiments.

[13] We used Amazon's Mechanical Turk (MT) for a pilot study. For the main experiment we used volunteers recruited via social media with an interest in improving their probabilistic judgments. We found it hard to design an incentive scheme for MT such that the users would make a good-faith effort to do well (and not e.g. declaring 50% on each judgment) while not cheating by looking up Fermi answers.

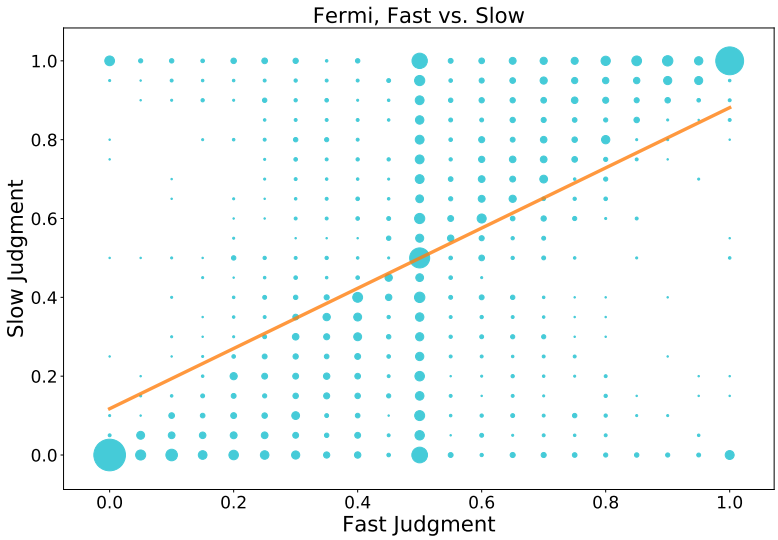
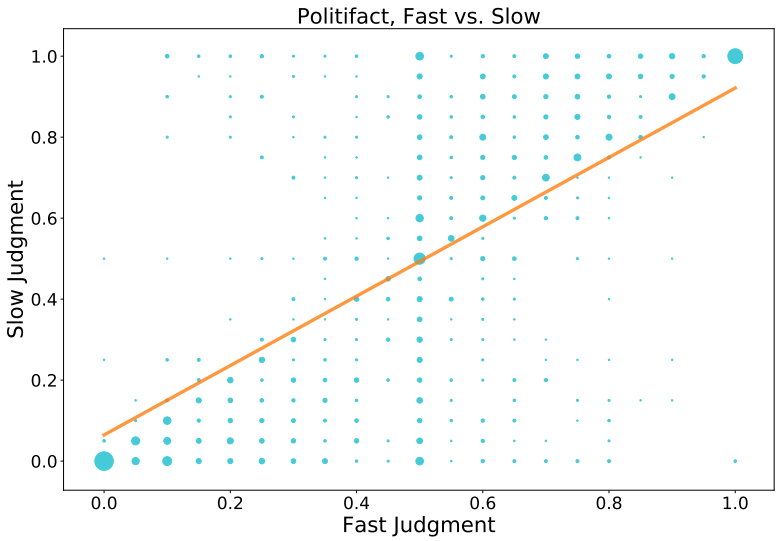


Figure 3: Correlation between fast and slow judgments for a given question-user pair for Fermi (above) and Politifact (below). Blue markers are proportional in size to number of instances and orange line is a linear regression through the data. Fast user judgments are often 50% but then become more confident in either direction given more time to think.



The Fermi data we collected consisted of 18,577 judgments, with a third of these being slow judgments. There were 619 distinct users and 2,379 Fermi estimation questions (with variable numbers of judgment per question). Each user answered 12.9 distinct questions on average (median 5). The Politifact dataset was smaller, containing 7,974 judgments, from 594 users, covering 1,769

statements. Each user answered 6.4 distinct questions on average (median 5).

2.4 Descriptive Statistics

Our goal was to use the dataset to train ML algorithms to predict slow judgments and such a dataset should have the following features:

1. *Large, clean, varied dataset*: as usual a large dataset with minimal noise is desirable. In particular there should be many judgments per user and many distinct questions. Possible sources of noise should be minimized (e.g. users are noisier when learning the task and if they pay less attention due to tedious or overly difficult questions).

2. *Fast-Slow Correlation*: fast judgments are informative about slow judgments but not too much. For some users and some questions, fast judgments may be similar to slow judgments. In other cases, the fast judgment will be uncertain or wrong.

3. *User Variation*: individual users vary in their overall performance and their performance in different kinds of question. We want to predict the slow judgments of an individual (not the ground-truth). In some tasks listed in Section 1.1 users vary because of different preferences. In Fermi and Politifact users might vary in their areas of knowledge (e.g. science vs. sport questions in Fermi).

4. *Question Variation*: individual questions vary in terms of how users answer them. For example, some questions are hard (producing random or systematically incorrect answers) and some are easy. This allows algorithms to predict user answers better based on features of the question.

Did our data have the features above? As noted in Section 2.3 the dataset was relatively small (especially Politifact) and many users only answered a small number of questions.

Correlations between fast and slow judgments are shown in Figure 3. They show that judgments started out uncertain but became more confident (in either direction) given time, and that confident fast judgments were less likely to change. Figure 4 shows that slow judgments were more accurate than fast on average. Nevertheless, the differences between slow, medium and fast judgments are not as large as we had hoped.

Did users vary in their accuracy? We expected users to vary in overall accuracy and accuracy for different kinds of question (e.g. some users do better on science than sports questions). However the level of variation among users was not high enough to rule out possibility that the best two-thirds of users did not vary (see Figure 5). This is a problem with the design of our experiment and we discuss it further in Section 4.

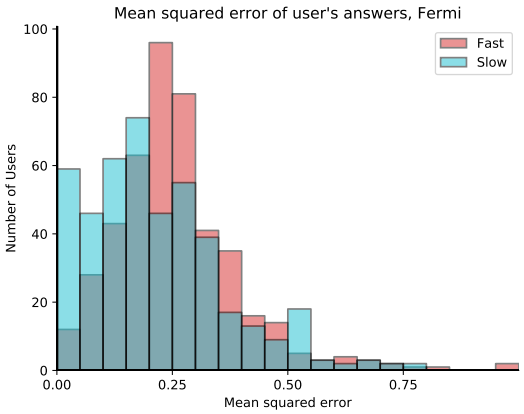


Figure 4: Variation in User Performance in Fermi for fast and slow judgments. Histogram shows number of users who obtained a particular level of performance (measured by MSE). Note that 0.25 is the MSE achieved by a user who always says 50%. While users have widely varying performance, much of this is due to noise and we can't rule out the possibility that most users do not vary in actual skill (see Figure 5).

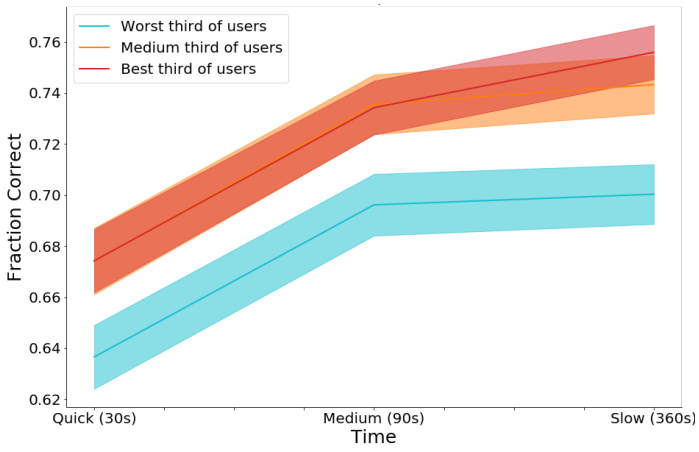


Figure 5: Mean performance on Fermi for different user quantiles. Users were divided into quantiles based on performance on a random half of the data and the figure shows their performance on the other half. Error bars show standard error from 5 random divisions into quantiles. The graph suggests we can't rule out the null hypothesis that the best two-thirds of users don't vary in their accuracy. (We removed users with less than k=6 judgments; the graphs were similar when k was set higher.)

Questions in both Fermi and Politifact did vary significantly in difficulty for human users (see Figure 6). However, there are less than 2500 questions for each task. This makes it hard for algorithms to generalize from the question text or meta-data without substantial pre-training.

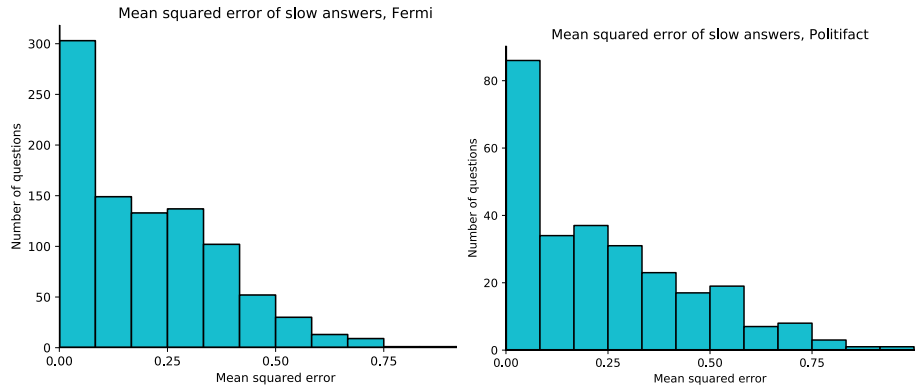


Figure 6: Variation in question difficulty for human users. Histograms show number of questions with a given MSE across users. Many questions had an MSE near zero (most users got them confidently right), while others were difficult (MSE near 0.25) or counter-intuitive (MSE greater than 0.25).

3 Predicting Slow Judgments with ML

This section presents a problem definition for predicting slow judgments using ML and results for baseline algorithms on our dataset. The ML task and algorithms are independent of the dataset and may be useful for future work in this area.

3.1 Task: Predicting Slow Judgments with ML

3.1.1 Task Definition

For both Fermi and Politifact the data consists of user probability judgments about questions. Each probability judgment $h(q, u, t) \in [0, 1]$ depends on the question text and meta-data features q, the user id u, and the time $t \in \{0, 1, 2\}$ (where the indices correspond to fast, medium, and slow judgments respectively).

We let $\hat{h}(q, u, t)$ be an algorithm's prediction of the user judgment $h(q, u, t)$. The task is to predict slow judgments $h(q, u, 2)$ from the test set. While only slow judgments appear in the test set, the training set consists of fast, medium and slow judgments.[14] The loss function is mean-squared error over all slow judgments in the test set. Let $T = \{(q, u)\}$ be the set of question-user pairs that appear in the test set. Then the objective is to minimize the mean squared error over slow judgments in the test set:

$$\frac{1}{|T|} \sum_{(q,u)\in T} \Big[h(q, u, 2) - \hat{h}(q, u, 2)\Big]^2 \tag{1}$$

Note that this problem setup is analogous to content-based collaborative filtering, where the task is to predict a user's held-out rating for an item (e.g. a movie), given the user id and a feature vector for the item [51].

[14]The test set is randomly sampled from the entire dataset and so the same users and questions can appear both in train and test.

3.1.2 Train on Fast Judgments, Predict Slow

Slow judgments are much more expensive to collect than fast judgments. Yet for questions that are either very easy or very difficult for a user, the fast and slow judgments will be very similar. So if a model can learn when fast judgments are a good proxy for slow judgments, it can predict slow judgments reasonably well from a much cheaper dataset (as discussed in Section 1.1).

In our dataset we have the user's slow judgment for a question whenever we have the fast judgment. Yet we can simulate the strategy of mostly collecting fast judgments by *masking* many of the slow judgments from our training set.[15] Our results are computed for three different levels of masking. "Unmasked" means that all slow judgments in the training set are included. "Heavy" means 90% of question-user pairs have the medium and slow removed and 7% have just the slow removed. "Light" means we removed slow and medium judgments for 60% of question-user pairs and removed just slow for 30%.[16]

3.2 Algorithms and Results

We ran the following baseline algorithms on our datasets:

1. *Collaborative Filtering (KNN and SVD)*: The prediction task is closely related to collaborative filtering (CF). The main difference is that instead of predicting a user's judgment about an item, we predict a user's judgment about a question at a given time (where we expect users to generally converge to zero or one given more time). To reduce our task to CF, we flatten the data by treating each user-time pair (u, t) as a distinct user. We applied the standard CF algorithms K-nearest-neighbors (KNN CF) and singular value decomposition (SVD CF), using the implementation in the Surprise library [52].

2. *Neural Collaborative Filtering*: We adapt the Neural CF algorithm [53] to our task. A neural net is used to map the latent question and user embeddings to a sequence of judgments for each time. *Linear Neural CF* forces the judgments to change linearly over time.

3. *Hierarchical Bayesian Linear Regression*: To predict the user judgment for a question we pool all user judgments for the same question (ignoring user identity) and regress using a Bayesian linear regression model. This model exploits the temporal structure of judgments but it discards the question features and user identity.

4. *Clairvoyant Model*: Since user slow judgments are correlated with the ground-truth, algorithms will do better on new questions to the extent they can predict the ground-truth. Predicting ground-truth without side-information is difficulty for Fermi and Politifact. However, we can investigate how well a model would do if it had access to the ground-truth. The

[15] This is like the standard practice in semi-supervised learning, where researchers remove all but 5% of the labels from the training set to simulate a situation where most data is unlabeled

[16] In order to achieve clear-cut separation of the training data and the held-out test set we also implemented a masking procedure before doing the additional masking for Light and Heavy. For each judgment in the test set, we removed the medium judgment made by the same user about the same question from the training set. We also stochastically remove the corresponding fast judgment with 80% probability.

"Clairvoyant" model simply predicts that each user will respond with the ground-truth answer with full confidence. The "Clairvoyant Mean" model predicts the base-rate probability for a user given the ground-truth of a question.

3.3 Results

Table 1 shows results for Fermi and Politifact with different levels of masking of slow judgments. Hierarchical Linear Regression and SVD perform best. Since Hierarchical Linear regression ignores both question features and user identity, this suggests it is difficult for algorithms to improve predictions by modeling questions and users. This is additional evidence that our dataset is problematic.

We had expected the Neural CF algorithms would do well, as they learn latent representations for both questions and users and (unlike SVD and KNN) they do not discard the temporal structure in the data. However, their poor performance is partly explained by the difficulty (discussed in Section 2.4) of distinguishing user variation from noise.[17]

The performance of Clairvoyant Mean suggests that non-clairvoyant algorithms could be improved with a strong language and reasoning model that accurately predicts the ground-truth.[18]

Table 1: Mean squared test error for various algorithms and different levels of masking slow judgments at training time. Note that Clairvoyant models had access to ground-truth and other models did not.

	Politifact			*Fermi*		
	Unmasked	Light	Heavy	Unmasked	Light	Heavy
KNN CF	0.127	0.130	0.133	0.115	0.125	0.134
SVD CF	0.124	**0.126**	**0.127**	0.102	0.113	**0.112**
Linear Neural CF	0.131	0.135	0.135	0.137	0.141	0.141
Neural CF	0.130	0.131	0.129	0.136	0.136	0.138
Hierarchical Lin. Reg.	**0.123**	**0.126**	**0.127**	**0.098**	**0.107**	0.114
Clairvoyant	0.242	0.242	0.242	0.216	0.216	0.216
Clairvoyant Mean	0.112	0.112	0.112	0.111	0.111	0.111
Always Guess 50%	0.137	0.137	0.137	0.138	0.138	0.138

4 Discussion: Mitigating Dataset Problems

Some problems with our dataset were described in Section 2.4. How could these problems be mitigated in future work?

Problem: Fast and slow judgments were too similar

There weren't big differences between fast and slow judgments. This can be addressed by choosing a task that requires more thinking and research than

[17] Like Hierarchical Linear Regression, the Linear Neural CF assumes that judgments evolve linearly over time. However, Linear Neural CF does not build in which data-points to regress on and wasn't able to learn this.

[18] We experimented with various language models (results not shown). We found it difficult to train or fine-tune these models on our small dataset without overfitting.

the Fermi and Politifact tasks. The task should also be *incremental*, such that a small amount of additional work yields a slightly better answer.[19] If additional work is unlikely to help, users will be averse to doing it (without big compensation).

Problem: Variation between users was hard to distinguish from noise

Lack of discernible variation can be addressed by collecting substantially more data per user and by trying to reduce noise (e.g. having practice questions, having less ambiguous questions, having tasks for which users are consistently fully engaged).

A related issue is that human responses were low in information content. Users assigned probabilities (from 20 discrete options) to binary ground-truth outcomes. Many questions were easy and were answered with 95-100% confidence, while others were very difficult and answered with 50-55% confidence. Furthermore, users were rarely anti-correlated with the ground truth, so slow judgments for true statements were generally answered somewhere in the range 60-100%, which is a small region of the response space in terms of mean squared error. This problem of low information content could be addressed by having a task where user responses are richer in information: e.g. scalar valued, categorical (with many categories), or a text response such as a movie review. The cost of this change is that the ML modeling task becomes harder.

Another way to address this issue is to have a task in which the goal for users is something other than guessing the ground-truth. There are many situations in which people make slow judgments for questions that are not about objective matters of fact. For example, when someone reviews a non-fiction book for their blog, they consider both objective qualities of the book ("Does it get the facts right?") and also subjective qualities ("Did I personally learn things from it? Will it help me make decisions?"). As noted above, there is nothing about our task definition or modeling that assumes the questions have an objective ground-truth.[20]

Problem: Question features were hard for algorithms to exploit

The questions in our tasks varied substantially in difficulty and content. Our models couldn't really exploit this variation, probably because (a) there were less than 2500 questions, and (b) predicting whether a question is challenging for humans is an intrinsically difficult task (drawing on NLP and world knowledge).

Having models that can make use of question and meta-data features is an important goal for our research. If models can only predict slow judgments based on other human judgments, they will not be able to generalize to new questions that no humans have yet answered.

The most obvious fix for this problem is to collect a much larger dataset. If the dataset contained millions of questions then language models would be better at learning to recognize easy vs. difficult questions. While collecting

[19] This would also allow a series of intermediate times between fast and slow.

[20] We collected data for a third task (not discussed elsewhere in this report), where the aim was to evaluate Machine Learning papers. We asked researchers to judge papers subjectively ("How useful is this paper for your research?") rather than objectively ("Should the paper be accepted for a conference?"). Unfortunately we did not a sufficient amount of data from volunteers. But we think some kind of variant on this task would be worth doing.

such a big dataset would be expensive, the cost could be mitigated by mainly collecting fast judgments.[21] An alternative to collecting a large dataset is to choose some object task (instead of Fermi and Politifact) for which pre-trained language models are useful (e.g. something related to sentiment analysis rather than judging the ground-truth of statements).

5 Conclusion

Machine Learning will only be able to automate a range of important real-world tasks (see Section 1.1) if algorithms get better at predicting human deliberative judgments in varied contexts. Such tasks are challenging due to AI-completeness and the difficulty and cost of data. We tried to create a dataset for evaluating ML algorithms on predicting slow judgments. The previous section discussed problems with our dataset and potential remedies. It's also worth considering alternative approaches to the challenge we outline in Section 1.1. First, some tasks may be more fruitful to model than Fermi estimation and political fact checking. Second, deliberation can be modeled explicitly by building on recent deep learning approaches [28, 29, 30, 31, 32, 33, 34, 35, 36, 37]. To more precisely capture how humans deliberate, we could also record the individual steps people take during deliberating (e.g. by requiring users to make their reasoning explicit or by recording their use of web search). Finally, we acknowledge that while predicting slow judgments is an important task for the future of AI, it may be difficult to make progress on today.

[21] It remains to be seen how well this can work if the goal is to predict slow judgments.

6 Acknowledgements

This work was supported by Future of Life Institute grant 2015-144846 (OE, AS). Important early work was done by Neal Jean and Girish Sastry. We thank Paul Christiano, David Krueger, and Tim Rocktäschel for valuable conversations. We are extremely grateful to everyone who volunteered and contributed data to our project via the site: thinkagain.ought.org.

References

[1] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. *arXiv preprint arXiv:1709.01507*, 2017.

[2] Wayne Xiong, Jasha Droppo, Xuedong Huang, Frank Seide, Mike Seltzer, Andreas Stolcke, Dong Yu, and Geoffrey Zweig. The microsoft 2016 conversational speech recognition system. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 5255–5259. IEEE, 2017.

[3] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerry-Ryan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. *arXiv preprint arXiv:1712.05884*, 2017.

[4] Jeremy Howard and Sebastian Ruder. Fine-tuned language models for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[5] Andrew Ng. Nuts and bolts of building ai applications using deep learning. NIPS, 2016.

[6] Daniel Kahneman and Patrick Egan. *Thinking, fast and slow*, volume 1. Farrar, Straus and Giroux New York, 2011.

[7] Gerd Gigerenzer, Peter M Todd, the ABC Research Group, et al. *Simple heuristics that make us smart*. Oxford University Press, 1999.

[8] Three ways machine learning is improving the hiring process. https://www.forbes.com/sites/forbestechcouncil/2018/03/26/three-ways-machine-learning-is-improving-the-hiring-process/#4d9518c290e8. Accessed: 2018-06-18.

[9] Fake news challenge. http://www.fakenewschallenge.org/. Accessed: 2018-06-18.

[10] Arxiv sanity preserver. http://arxiv-sanity.com. Accessed: 2018-06-18.

[11] Wikimedia research: Detox. https://meta.wikimedia.org/wiki/Research:Detox. Accessed: 2018-06-18.

[12] It's easy to slip toxic language past alphabet's toxic-comment detector. https://www.technologyreview.com/s/603735/its-easy-to-slip-toxic-language-past-alphabets-toxic-comment-detector/. Accessed: 2018-06-18.

[13] Jigsaw toxic comment classification challenge. https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge. Accessed: 2018-06-18.

[14] William Yang Wang. "Liar, Liar Pants on Fire": A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*, 2017.

[15] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 191–198. ACM, 2016.

[16] Wikipedia: AI-complete. https://en.wikipedia.org/wiki/AI-complete. Accessed: 2018-06-18.

[17] Geoffrey Irving, Paul Christiano, and Dario Amodei. AI safety via debate. *arXiv preprint arXiv:1805.00899*, 2018.

[18] Nick Bostrom. *Superintelligence: Paths, Dangers, Strategies*. Oxford University Press, Inc., New York, NY, USA, 1st edition, 2014.

[19] Paul Christiano. AI Alignment Blog. https://ai-alignment.com. Accessed: 2018-06-18.

[20] William Saunders, Girish Sastry, Andreas Stuhlmueller, and Owain Evans. Trial without error: Towards safe reinforcement learning via human intervention. *arXiv preprint arXiv:1707.05173*, 2017.

[21] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, pages 4302–4310, 2017.

[22] Paul Christiano. Act-based agents. https://ai-alignment.com/act-based-agents-8ec926c79e9c. Accessed: 2018-06-18.

[23] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, pages 4565–4573, 2016.

[24] Owain Evans, Andreas Stuhlmüller, and Noah D Goodman. Learning the preferences of bounded agents. NIPS Workshop on Bounded Optimality, 2015.

[25] Owain Evans, Andreas Stuhlmüller, and Noah D Goodman. Learning the preferences of ignorant, inconsistent agents. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 323–329. AAAI Press, 2016.

[26] Paul Christiano. Approval maximizing representations. https://ai-alignment.com/approval-maximizing-representations-56ee6a6a1fe6. Accessed: 2018-06-18.

[27] Paul Christiano and Ajeya Cotra. Iterated distillation and amplification. https://ai-alignment.com/iterated-distillation-and-amplification-157debfd1616. Accessed: 2018-06-18.

[28] Drew A Hudson and Christopher D Manning. Compositional attention networks for machine reasoning. *arXiv preprint arXiv:1803.03067*, 2018.

[29] Richard Evans, David Saxton, David Amos, Pushmeet Kohli, and Edward Grefenstette. Can neural networks understand logical entailment? *arXiv preprint arXiv:1802.08535*, 2018.

[30] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40, 2017.

[31] Richard Evans and Edward Grefenstette. Learning explanatory rules from noisy data. *Journal of Artificial Intelligence Research*, 61:1–64, 2018.

[32] Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*, 2015.

[33] Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, et al. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471, 2016.

[34] Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. Towards AI-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*, 2015.

[35] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. In *International Conference on Machine Learning*, pages 1378–1387, 2016.

[36] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015.

[37] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638, 2016.

[38] Cezary Kaliszyk, François Chollet, and Christian Szegedy. Holstep: A machine learning dataset for higher-order logic theorem proving. *arXiv preprint arXiv:1703.00426*, 2017.

[39] Yue Shi, Martha Larson, and Alan Hanjalic. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys (CSUR)*, 47(1):3, 2014.

[40] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006). *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

[41] Avital Oliver, Augustus Odena, Colin Raffel, Ekin D Cubuk, and Ian J Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. *arXiv preprint arXiv:1804.09170*, 2018.

[42] Jan Deriu, Maurice Gonzenbach, Fatih Uzdilli, Aurelien Lucchi, Valeria De Luca, and Martin Jaggi. Swisscheese at semeval-2016 task 4: Sentiment classification using an ensemble of convolutional neural networks with distant supervision. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, number EPFL-CONF-229234, pages 1124–1128, 2016.

[43] David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. *arXiv preprint arXiv:1511.03643*, 2015.

[44] Jessica Taylor, Eliezer Yudkowsky, Patrick LaVictoire, and Andrew Critch. Alignment for advanced machine learning systems. *Machine Intelligence Research Institute*, 2016.

[45] Volodymyr Kuleshov and Stefano Ermon. Estimating uncertainty online against an adversary. In *AAAI*, pages 2110–2116, 2017.

[46] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6405–6416, 2017.

[47] Sentiment classification on large movie review. https://www.kaggle.com/c/sentiment-classification-on-large-movie-review. Accessed: 2018-06-18.

[48] Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. A dataset of peer reviews (peerread): Collection, insights and nlp applications. *arXiv preprint arXiv:1804.09635*, 2018.

[49] Sara Klingenstein, Tim Hitchcock, and Simon DeDeo. The civilizing process in london's old bailey. *Proceedings of the National Academy of Sciences*, 111(26):9419–9424, 2014.

[50] Gamaleldin F Elsayed, Shreya Shankar, Brian Cheung, Nicolas Papernot, Alex Kurakin, Ian Goodfellow, and Jascha Sohl-Dickstein. Adversarial examples that fool both human and computer vision. *arXiv preprint arXiv:1802.08195*, 2018.

[51] Oren Barkan, Noam Koenigstein, and Eylon Yogev. The deep journey from content to collaborative filtering. *arXiv preprint arXiv:1611.00384*, 2016.

[52] Nicolas Hug. Surprise, a Python library for recommender systems. http://surpriselib.com, 2017.

[53] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 173–182, Republic and Canton of Geneva, Switzerland, 2017. International World Wide Web Conferences Steering Committee.