Bulletin of the Technical Committee on

Data Engineering

December 2019 Vol. 42 No. 4

IEEE Computer Society

Letters

Opinions

Special Issue on Human Powered AI Systems

Conference and Journal Notices

The TC on Data Engineering

Membership in the TC on Data Engineering is open to all current members of the IEEE Computer Society who are interested in database systems. The TCDE web page is http://tab.computer.org/tcde/index.html.

The Data Engineering Bulletin

The Bulletin of the Technical Committee on Data Engineering is published quarterly and is distributed to all TC members. Its scope includes the design, implementation, modelling, theory and application of database systems and their technology.

Letters, conference information, and news should be sent to the Editor-in-Chief. Papers for each issue are solicited by and should be sent to the Associate Editor responsible for the issue.

Opinions expressed in contributions are those of the authors and do not necessarily reflect the positions of the TC on Data Engineering, the IEEE Computer Society, or the authors' organizations.

The Data Engineering Bulletin web site is at
http://tab.computer.org/tcde/bull_about.html.

Letter from the Editor-in-Chief

One of the beauties of the Data Engineering Bulletin, with a history of 43 years and 157 issues, is that it chronicles how topics of database research evolve and sometimes reinvent themselves over time. Phil Bernstein's opinion piece in this issue, titled "Resurrecting Middle-Tier Distributed Transactions," is another testimony to this beauty. Bernstein tells an interesting story of transaction processing monitors running on middle-tier servers, and predicts the return of middle-tier distributed transactions to the mainstream after a 15-year decline.

Guoliang Li put together the current issue consisting of 6 papers on the interactions between database systems and AI. This is a fascinating topic. Traditional databases are heavily optimized monolithic systems designed with heuristics and assumptions. But recent work has shown that critical data structures such as database indices are merely models, and can be replaced with more flexible, faster, and smaller machine learned models such as neural networks. This opens the door to using data driven approaches for system design. On the other hand, deep learning is still facing the challenge in incorporating database accesses in end-to-end training, which hampers the use of existing structured knowledge in learning.

Haixun Wang
WeWork Corporation

Letter from the Special Issue Editor

The fast development of AI technology has changed our dail life significantly, from personal assistant such as Siri to self-driving cars, online recommendations sites such as Netflix and Amazon and many other real-world AI applications have emerged and brought great benefits to humanity. All these AI-powered applications have shown great intelligence if their models are well trained. For example, Alpha go, a well-trained go player defeated top human players in the world. However, the current AI technology relies strongly on the quality of the training datasets and can be restricted by the cognitive nature of a task. Tasks such as recognizing objects from blurred images and understanding sentiment from complicated and poorly-structured sentences still need human assistance. Therefore, in this issue, we study an interesting topic, human-powered AI. Compared to current AI technology which is more advanced in solving closed domain problems, human intelligence is more advanced in addressing open domain problems, such as arts and design. How to seamlessly incorporate human intelligence into the whole process of AI is the theme of human-powered AI.

In this issue, we present works on human-powered AI from different aspects,

The first paper discusses a challenge and essential problem in Human-powered AI, how to divide computation between human and AI effectively to achieve a specific target. Based on the discussion, the authors have introduced a set of dimensions and terms to classify existing solutions on this topic.

The second paper addresses the challenges on how to untilize human intelligence (crowdsourcing) on federate learning, which is one of the popular solutions to overcome the problem of data isolation and data privacy in a distributed learning environment.

The third paper addresses the challenges on providing an end to end solution in human-in-the-loop of machine learning, from data extract, data integration, data cleaning, data labelling to machine learning and inference.

The fourth paper presents human-powered AI from the angle of AI technology, specifically, how to use AI technology to model, discover and explore human behaviour for human intelligence data management and mining.

The last paper presents a real application of human-powered AI techniques for online misinformation detection, challenges on various aspects in implementing such a system are outlined.

We would like to thank all the authors for their insightful contributions.

Lei Chen
Hong Kong University of Science and Technology

Resurrecting Middle-Tier Distributed Transactions

Philip A. Bernstein
Microsoft Research, Redmond, WA 98052

1 Introduction

Over the years, platforms and application requirements change. As they do, technologies come, go, and return again as the preferred solution to certain system problems. In each of its incarnations, the technology's details change but the principles remain the same. One such technology is distributed transactions on middle-tier servers. Here, we argue that after a 15-year decline, they need to return to the mainstream.

In the 1980's, Transaction Processing (TP) monitors were a popular category of middleware product that enabled customers to build scalable distributed systems to run transactions. Example products were CICS (IBM), Tuxedo (AT&T for Unix), ACMS (DEC for VAX/VMS), and Pathway (Tandem for Guardian) [4]. Their main features were multithreaded processes (not supported natively by most operating systems), inter-process communication (usually a crude form of remote procedure call), and a forms manager (for end users to submit transaction requests). The TP monitor ran on middle-tier servers that received transaction requests from front-end processors that communicated with end-user devices, such as terminals and PC's, and with back end database servers. The top-level application code executed on the middle-tier and invoked stored procedures on the database server.

In those days, database management systems (DBMS's) supported ACID transactions, but hardly any of them supported distributed transactions. The TP monitor vendors saw this as a business opportunity and worked on adding a transaction manager feature that implemented the two-phase commit protocol (2PC). Such a feature required DBMS's to expose Start, Prepare, Commit, and Abort as operations that could be invoked by the TP monitor. Unfortunately, most of them didn't support Prepare, and even if they did, they didn't expose it to applications. They were willing to do so, but they didn't want to implement a different protocol for each TP monitor product. Thus, the XA standard was born, which defined TP monitor and DBMS interfaces (including Prepare) and protocols that allowed a TP monitor to run a distributed transaction across DBMS servers [17].

This middle-tier architecture for distributed transactions was popular for about 20 years, into the late 1990s. Then TP monitors were replaced by Application Servers, which integrated a TP monitor with web servers, so it could receive transaction requests over HTTP, rather than receiving them from devices connected by a local area or terminal network. Examples include Microsoft Transaction Server, later renamed COM+, and Java Enterprise Edition (JEE), implemented by IBM's WebSphere Application Server, Oracle's WebLogic Application Server, and Red Hat's JBoss Application Server [12]. The back end architecture was the same as before. Each transaction started executing on a middle-tier server and invoked stored procedures to read and write the database.

Although this execution model is still widely used, starting in the early 2000's it fell out of favor for new application development, especially for applications targeted for cloud computing. More database vendors offered built-in support for distributed transactions, so there was less need to control the distributed transaction from the middle tier. A larger part of database applications executed on data that was cached in the middle tier. And the NoSQL movement argued that distributed transactions were too slow, that they limited scalability, and that customers rarely needed them anyway [11]. Eventual consistency became all the rage [18].

The critics of distributed transactions had some good points. But in the end, developers found that mainstream

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

programmers really did need ACID transactions to make their applications reliable in the face of concurrent access to shared data and despite server failures. Thus, some NoSQL (key-value) stores added transaction support (e.g., CosmosDB [2], DynamoDB [8]). Google, which had initially avoided support for multi-row distributed transactions in Bigtable [5], later introduced them in Spanner [6]. There are now many cloud storage services and database products that support distributed transactions.

Like product developers, database researchers have also focused on distributed transactions for back-end database systems. Almost universally, they assume that transactions execute as stored procedures and that middle-tier applications invoke those stored procedures but do not execute the transaction logic themselves.

2 Stateful Middle-Tier Applications

This focus on stored procedures is well justified by the needs of traditional TP applications. However, stored procedures are not a good way of encapsulating application logic for a growing fraction of stateful applications that run on the middle tier. These include multi-player games, datacenter telemetry, Internet of Things, and social and mobile applications. Objects are a natural model for the entities in these applications, such as games, players, datacenter hardware, sensors, cameras, customers, and smart phones. Such applications have a large number of long-lived stateful objects that are spread over many servers and communicate via message passing. Like most new applications, these applications are usually developed to run on cloud computing platforms.

These applications typically execute on middle-tier servers, rather than as stored procedures in database servers. They do this for many reasons. They need large main memory for the state of long-lived objects. They often have heavy computation needs, such as rendering images or computing over large graphs. They use a lot of computation for message passing between objects so they can scale out. And they need computation to be elastic, independent of storage requirements. These needs are satisfied by compute servers that are cheaper than database servers because they have less storage. Hence, these apps run on compute servers in the middle tier.

2.1 Requirements for Mid-Tier Cloud Transactions

Some middle-tier applications need transactions because they have functions that read and write the state of two or more stateful objects. For example, a game may allow users to buy and sell virtual game objects, such as weapons, shields, and vehicles. A telemetry application may need to process an event exactly once by removing it from a queue and updating telemetry objects based on that event. A social application may need to add a user to a group and modify the user's state to indicate membership in that group. Each of these cases needs an ACID transaction over two or more objects, which may be distributed on different servers. Since these applications are usually developed to run on cloud computing platforms, distributed transaction support must be built into the cloud platform, a capability that is rarely supported today for cloud computing.

Distributed transactions for middle-tier applications on a cloud computing platform have four requirements that differ from those supported by the late-1990's products that run transactions on the middle-tier. First, like all previous transaction mechanisms, they need to offer excellent performance. But unlike previous mechanisms, it's essential that they be able to scale out to a large number of servers, leading to the first requirement: The system must have high throughput and low transaction latency, at least when transactions have low contention, and in addition must scale out to many servers.

To scale computation independently of storage, these applications typically save their state in cloud storage. The developers' choice of cloud storage service depends on their application's requirements (e.g., records, documents, blobs, SQL), their platform provider's offerings (e.g., AWS, Azure, Google), their employer's storage standards, and their developers' expertise. Thus, we have this second requirement: The transaction mechanism must support applications that use any cloud storage service.

The transaction mechanism needs persistent storage to track transaction state: started, prepared, committed, or aborted. Like the apps themselves, it needs to use cloud storage for this purpose, which is the third requirement: The transaction mechanism must be able to use any of the cloud storage services used by applications.

The traditional data structure for storing transaction state is a log. The transaction manager relies on the order of records in the log to understand the order in which transactions executed. Although cloud vendors implement logs to support their database services, they do not expose database-style logging as a service for customers, leading to a fourth requirement: The transaction mechanism cannot rely on a shared log, unless it implements the log itself, in which case the log must run on a wide variety of storage services.

Due to the latency of cloud storage, requirements (2)-(4) create challenges in satisfying requirement (1).

The above requirements are a first cut, based on today's applications and platforms. It is also worth targeting variations. For example, requirement (1) could include cost/performance, which might require a tradeoff against scalability. And (4) might go away entirely if cloud platforms offer high-performance logging as a service.

3 An Implementation in the Orleans Framework

The rest of this paper sketches a distributed transaction mechanism that satisfies the above requirements [9]. Our group built it for Microsoft's actor-oriented programming framework, called Orleans, which is open source and runs on both Windows and Linux [16]. The distributed transaction project is part of a longer-term effort to enrich Orleans with other database features to evolve it into an actor-oriented database system that supports geo-distribution, stream processing, indexing, and other database features [3].

3.1 Two-Phase Commit and Locking

For ACID semantics, Orleans transactions use two-phase commit (2PC) and two-phase locking (2PL). Our first challenge was to obtain high throughput and scalability despite the requirement to use cloud storage. In our runs, a write to cloud storage within a datacenter takes 20 ms and has high variance. With 2PC, a transaction does two synchronous writes to storage. Therefore, if 2PL is used, a transaction holds locks for 40ms, which limits throughput to 25 transactions/second (TPS). Low-latency SSD-based cloud storage is faster, but still incurs over 10 ms latency, plus higher cost. To avoid this problem, we extended early lock release to 2PC [1, 7, 10, 13, 14, 15]. After a transaction T1 terminates, it releases locks before writing to storage in phase one of 2PC. This allows a later transaction T2 to read/update T1's updated objects. Thus, while T1 is writing to storage, a sequence of later transactions can update an object, terminate, and then unlock the object. To avoid inconsistency, the system delays committing transactions that directly or indirectly read or overwrite T1's writeset until after T1 commits. And if T1 aborts, then those later dependent transactions abort too. Using this mechanism, we have seen transaction throughput up to 20x that of strict 2PL/2PC.

3.2 Logging

Our initial implementation used a centralized transaction manager (TM) per server cluster [9]. It ran on an independent server and was multithreaded. Since message-passing is a potential bottleneck, it batched its messages to transaction servers. It worked well with throughput up to 100K TPS. However, it had three disadvantages: it was an obvious bottleneck for higher transaction rates; a minimum configuration required two servers (i.e., primary and backup TM) in addition to servers that execute the application; and it added configuration complexity since TM servers did not run Orleans and thus had to be deployed separately from application servers.

These disadvantages led us to redesign the system to avoid a centralized TM. Instead, we embed a TM in each application object. Each TM's log is piggybacked on its object's storage. This TM-per-object design avoids the above disadvantages and improves transaction latency by avoiding roundtrips to a centralized TM. However, it doesn't work for objects that have no updatable storage. For example, an object that performs a money transfer calls two stateful objects, the source and target of the transfer, but it has no state itself. We allow such an object to participate in a transaction by delegating its TM function to a stateful participant in the transaction, that is, one that has updatable storage.

Orleans transactions write object state to a log to enable undo when a transaction aborts. This is impractical for large objects and is a poor fit for concurrency control that exploits operation commutativity. We therefore developed a prototype that logs operations.

4 Summary

Many new cloud applications run their logic on the middle tier, not as stored procedures. They need distributed transactions. Thus, cloud computing platforms can and should offer scalable distributed transactions.

5 Acknowledgments

I'm grateful to the team that built the initial and final implementations of Orleans transactions: Jason Bragg, Sebastian Burckhardt, Tamer Eldeeb, Reuben Bond, Sergey Bykov, Christopher Meiklejohn, Alejandro Tomsic, and Xiao Zeng. I also thank Bailu Ding and Dave Lomet for suggesting many improvements to this paper.

References

[1] Athanassoulis, Manos ; Johnson, Ryan ; Ailamaki, Anastasia ; Stoica, Radu, Improving OLTP Concurrency through Early Lock Release, EPFL-REPORT-152158, https://infoscience.epfl.ch/record/152158?ln=en, 2009.

[2] Azure CosmosDB, https://azure.microsoft.com/en-us/services/cosmos-db/

[3] Bernstein, P.A., M., T. Kiefer, D. Maier: Indexing in an Actor-Oriented Database. CIDR 2017

[4] Bernstein, P. A., E. Newcomer: Chapter 10: Transactional Middleware Products and Standards, in Principles of Transaction Processing, Morgan Kaufmann, 2nd ed., 2009.

[5] Chang, F., J. Dean, S. Ghemawat, W.C. Hsieh, D.A. Wallach, M. Burrows, T. Chandra, A. Fikes, R.E. Gruber: Bigtable: A Distributed Storage System for Structured Data. ACM Trans. Comput. Syst. 26(2): 4:1-4:26 (2008)

[6] Corbett, J.C. et al: Spanner: Google's Globally Distributed Database. ACM Trans. Comput. Syst. 31(3): 8:1-8:22 (2013)

[7] DeWitt, D.J., R.H. Katz, F. Olken, L.D. Shapiro, M. Stonebraker, D.A. Wood: Implementation Techniques for Main Memory Database Systems. SIGMOD 1984: 1-8

[8] DynamoDB, https://aws.amazon.com/dynamodb/

[9] Eldeeb, T. and P. Bernstein: Transactions for Distributed Actors in the Cloud. Microsoft Research Tech Report MSR-TR-2016-1001.

[10] Graefe, G., M. Lillibridge, H. A. Kuno, J. Tucek, A.C. Veitch: Controlled lock violation. SIGMOD 2013: 85-96

[11] Helland, P., Life beyond Distributed Transactions: an Apostate's Opinion. CIDR 2007: 132-141

[12] Java EE documentation, http://www.oracle.com/technetwork/?java/javaee/documentation/index.html

[13] Larson, P-A, et al.: High-Perf. Concurrency Control Mechanisms for Main-Memory Databases. PVLDB 2011

[14] Levandoski, L.J., D.B. Lomet, S. Sengupta, R. Stutsman, R. Wang: High Performance Transactions in Deuteronomy. CIDR 2015

[15] David B. Lomet: Using Timestamping to Optimize Two Phase Commit. PDIS 1993: 48-55

[16] Orleans, http://dotnet.github.io/orleans

[17] The Open Group, Distributed Transaction Processing: The XA Specification, http://pubs.opengroup.org/onlinepubs/009680699/toc.pdf.

[18] Vogels W., Eventually Consistent. ACM Queue 6(6): 14-19 (2008)

Computational Division of Labor with Human and AI Workers

Atsuyuki Morishima, Masaki Matsubara, Kei Wakabayashi, Nobutaka Suzuki, Hiroyoshi Ito
University of Tsukuba
{mori, masaki, kwakaba, nsuzuki, ito}@slis.tsukuba.ac.jp

Abstract

Computational division of labor addresses the design and analysis of algorithms for division of labor problems and will be one of the key issues in Future of Work. The problems deal with interactions among multiple worker and task classes in task decomposition, worker recruitment and education, and task assignments to AI and humans, for efficiently completing given tasks. We survey some of the related literature and discuss challenges and open problems.

1 Introduction

Division of Labor is known as an essential factor for achieving wealth of human beings. In division of labor, individuals and organizations acquire specialised capabilities and play different roles in a system, and either form combinations or trade to take advantage of their and others' capabilities.

In the recent years, more and more information on tasks is being circulated on the Internet, and there are many labor resources accessible through it. This allows us to take the computational approach to division of labor, in which AI and algorithm agents assign appropriate tasks to workers to achieve specific goals. There are many studies that address topics related to division of labor. For example, many papers on crowdsourcing address algorithms for task assignment considering given objective functions, such as price and required time. However, division of labor is different from just "decomposing into microtasks." Task decomposition itself does not necessary cause division of labor, if we obtain a set of tasks each of which requires workers to have the same set of capabilities represented by their attributes (such as skills and locations of workers). We need *task classes* that require different capabilities and *worker classes* that have workers with different capabilities. For example, assume that we have a set of sentences in English that needs to be translated into Braille in Japanese. Decomposing into a set of tasks each of which translates one sentence into Braille does not cause a division of labor, because we have only one task class that require the same skills. In contrast, decomposing them into a set of translation tasks from English to Japanese and another set of translation tasks from Japanese into Japanese Braille result in division of labor; then we have two task classes that require different specialized capabilities for translation. In addition, we may be able to further decompose the obtained task classes. For example, we can find a *subclass* under the English-Japanese translation task class in which the sentences require detailed knowledge of a particular domain (e.g., Japanese pop stars). Then, workers who have the knowledge can be assigned to the tasks in the subclass. Note that how to decompose task classes depends on the availability of workers in the

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

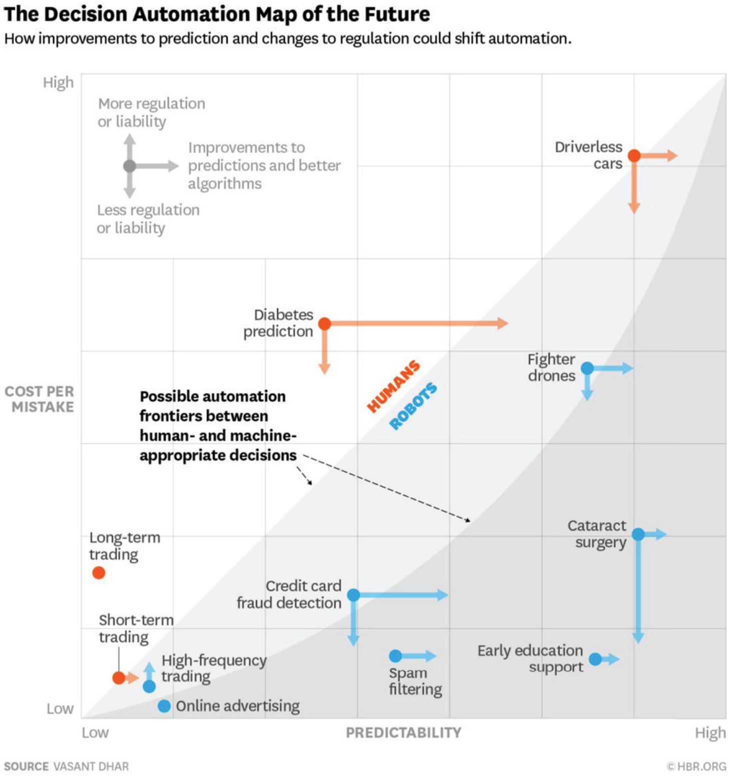


Figure 1: Whether decisions in particular areas should be made by humans or AI [4]. The x-axis is the predictability by AI. The y-axis is the cost per mistake. This figure states that decisions can be made by AI if it has high predictability and low cost per mistake. We already have many things for which we can use AI (dots) and we expect to have more things as we have improvement in predictions and algorithms (horizontal arrows), and regulations or liability changes (vertical arrows) in the near future.

worker pool. If we had a lot of people who are good at translating English Braille into Japanese Braille, we would have different subclasses: Translating English into Braille and translating it into Japanese one.

Adam Smith pointed out that the efficiency of division of labor comes from the following benefits: Increase of workers' capabilities, lower switching cost of tasks, and machines taking place of manual labor [27]. Taking different roles and trading or combining their abilities and products has a dramatic effect; it would be impossible for us to obtain many things today, let's say a smartphone, in exchange for the work for several to tens of hours, if each of us made our smartphone ourselves, without having a large amount of people who produce materials and semiconductor chips, design the electronic circuit and user interface, develop software, etc. Computational division of labor will be one of the important issues in Future of Work [1]. However, as we will show, this area is still in its infancy and there are a lot of things we can do.

While we have an environment that makes us ready to pursue the computational division of labor today, there is the demand for the computational approach. We recently noticed that worker availability can be changed in a disease pandemic. However, one of the most important factors is the rapid growth of AI. Figure **??** shows whether decisions in particular areas should be performed by humans or AIs at present and the potential changes in the near future. It depends not only the predictability but also cost per mistake. As shown in the figure, we already have a lot of things we can rely on AI, and there will be more and more in the near future, as we see improvement to predictions and better algorithms, and regulations or liability changes. Many new kinds of commercial services that employ new combinations of AI and human resources are emerging at a great speed. This means that AI is important not only for implementing solutions of computational division of labor, but also as *AI workers*,

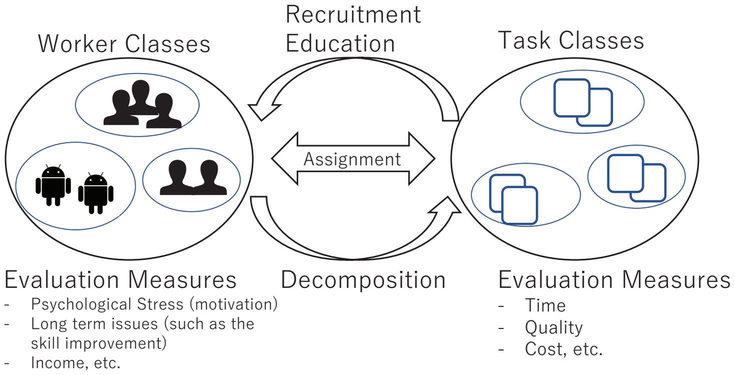


Figure 2: Framework for computational division of labor

that work in collaboration with human workers. In addition, the change spreads world wide in a short period of time; we see accelerating technology change [16] and new services and industries are deployed at a large scale internationally. Such a *scale* and *speed* of change caused by the *rapid growth of AI technology* requires the computational approach to the division of labor problems.

Note that there are a variety of tasks beyond decision tasks, which often require other capabilities than high predictability such as flexible response, some of which are difficult for the AI today, although the situation may change in the near future. For example, think about a conveyor belt sushi restaurant which is popular in Japan. Although making oval-shaped rices for sushi used to be considered as an expert task that is allowed to be performed by experienced staff members only, the advance of technology allowed machines to make the rices in conveyor belt sushi restaurants. Even fish slices to be put onto the rice are being made by machines with AI. However, there are many other tasks that are being performed by humans workers. They *develop* and *teach* AI, communicate with customers, cook special kinds of sushi, and deal with non-routine issues, all of which are difficult tasks for AI today.

In this article, we investigate problems on computational division of labor and look at the current status of research on crowdsourcing and human-in-the-loop systems on the Internet from the division-of-labor perspective. We chose these areas because in the near future, we expect that many jobs will be supplied by human-in-the-loop online job platforms [8].

Paper Outline. The rest of the paper is organized as follows. In Section 2, we explain what computational division of labor problems deal with and introduce a set of dimensions and terms to classify existing solutions for problems related to computational division of labor. In Section 3, we look into some of existing literature and investigate the current status. Section 4 discusses challenges and open problems.

2 Computational Division of Labor

The "computational division of labor" is not a new concept, and has been addressed in many related areas, notably in crowdsourcing research, although the state is still in its infancy as we will show in the next section.

Figure 1 shows the framework for computational division of labor. It has two key components: *task classes* and *worker classes*. A task class defines a set of task instances that require a specific set of capabilities in order to perform them. A worker class defines a set of workers having a specific set of capabilities. The task and worker classes have *mutual dependence*; i.e., good task decomposition results in

$$\underset{T\in\mathcal{T}}{\arg\max}\ \vec{f}(\mathcal{M}(W,T)),$$

where W is a set of workers, $\mathcal{T}$ is the collection of feasible task decompositions for W, $\mathcal{M}$ is the appropriate

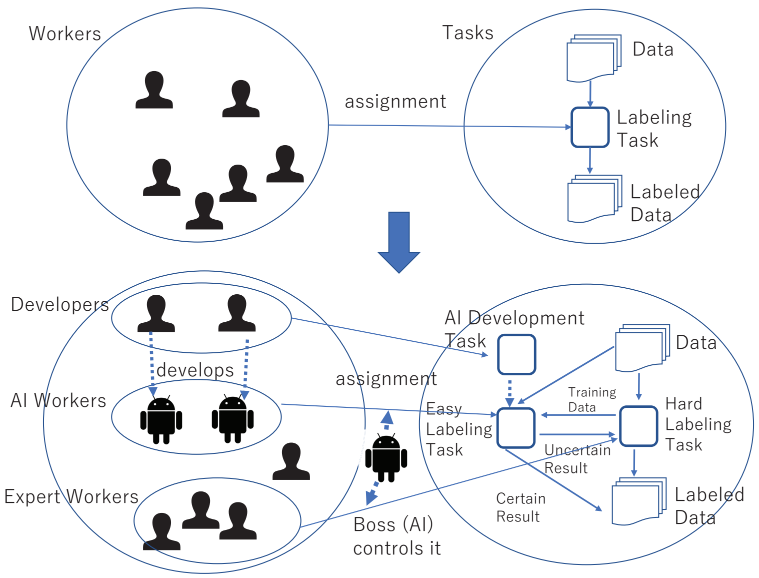


Figure 3: Example of dividing tasks and workers into multiple classes. If we know we have software developers in the worker pool, we can decompose the original task into a set of tasks consisting of AI development task, AI labeling task, and human labeling task. Since the tasks include tasks for AI workers, we recruit software developers so that AI workers are added to the worker pool.

assignment from tasks to qualified workers, and $\vec{f}$ is the multi objective functions that define what is good assignment. The functions may include a variety of things, including the result quality, the expected cost due to mistakes, and human factors related to workers [2]. In the opposite direction, what skills workers should obtain and what kinds of AI workers should be developed are described by

$$\mathop{\arg\max}_{W \in \mathcal{W}} \ \vec{f}(\mathcal{M}(W, T)),$$

where $\mathcal{W}$ is the collection of sets of workers augmented by T, i.e., each $W \in \mathcal{W}$ can be a set of workers with new or better skills, or an extended set of workers with newly developed AI workers. In addition, the set of skills the decomposed tasks require will affect how we should recruit workers.

Figure 3 gives an example process of computational division of labor for labeling tasks. If we know that we have software developers in the worker pool, we can decompose the original task into a set of tasks consisting of AI development tasks, easy labeling tasks (to be assigned to AI workers) and hard labeling tasks (assigned to human workers). Since the tasks include tasks for AI workers, we recruit software developers so that AI workers are added to the worker pool.

Assignment of workers to tasks is often determined by considering not only short-term benefits (such as time, quality, cost) for requesters, but long-term benefits, such as social sustainability and inclusion, with the three important benefits (skill improvement, low switching cost, and AI utilization) of division of labor considered.

The essential part of division of labor in the framework is that there must be a variety of task classes, each of which requires different expertise or abilities to complete them. Division of labor can lead to efficient society by exploiting the following advantages [27]:

Increase of Workers' Capabilities Some tasks may require special expertise and workers need to have experiences or be trained for doing a good job on the task. This sometimes requires long-term commitment of workers to a set of tasks that require particular capabilities.

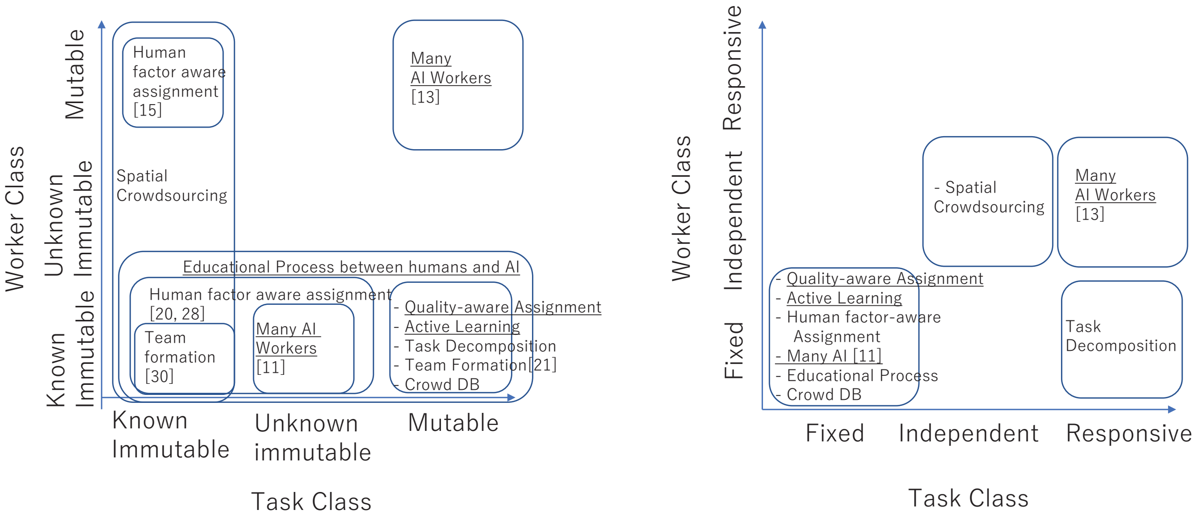


Figure 4: Two views of a solution space of computational division of labor and the current status of solutions in related topics. On the left is the view with instnace-level mutability class dimentions. On the right is one with class-level dynamicity dimensions. Underlined Topics deal with both of human and AI workers, while the others deal with human workers only.

Lower Switching Cost Cost for switching into completely new tasks is generally high for humans. Taking into this factor when assigning tasks to human workers increase the efficiency of manual labor.

Machine (AI) Taking Place of Manual Labor Dividing the task into sub-tasks can increase the opportunities for machines (AI workers) to do the task, if the task is appropriately extracted so that the AI worker is capable of doing the task. In order to achieve this, we need to address meta-level algorithms that explicitly deal with capabilities of available AI workers.

3 Related Research and Current Status

There is a lot of research done related to computational division of labor. This section tries to organize relevant topics especially in the area of human-in-the-loop database systems, crowdsourcing and machine learning. Although crowdsourcing research generally focuses on the case where workers are human workers and the objective functions are given in terms of each workflow (e.g., quality, time, money), there are papers that address important components of division of labor such as task and worker classes and the three benefits. Human-in-the-loop database systems combine human and AI workers in the way that human workers collect data that are not stored in the current snapshot of database. Regarding the division of labor between humans and AI workers, machine learning plays an essential role in some division of labor problems. Some studies such as supervised learning focus on addressing how to replace human labor by teaching AIs via a dynamic interaction between AIs and human workers.

3.1 Dimensions and Overview

Existing solutions for problems related to computational division of labor can be seen from the division of labor perspective by placing them in a space with the dimensions related to task and worker classes, such as follows:

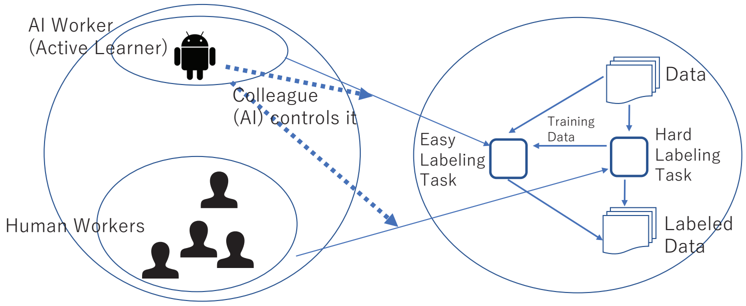


Figure 5: Active learning from the perspective of computational division of labor. The active learner is a colleague AI worker of human workers, who controls the task assignment.

A. Instance level mutability of Task and Worker Classes. The dimentions represents the interaction between task (or worker) classes and their instances with the three categories:

Known Immutable The solution assumes that we already know which class each task or worker belongs to, and the instance-of relationship does not change.

Unknown Immutable The solution does not know which class each task or worker belongs to at first, but the instance-of relationship does not change once it decides the membership.

Mutable The solution changes the instance-of relationship according to a change of situation. Thus the number of instances of each class changes.

For example, some research papers classify crowd workers into novice and expert workers. Other papers classify them into groups each of which contains similar workers in terms of accuracy per labels. If a method assumes that we know who are novice and expert workers in advance, it is labeled with "Known Immutable." If a method measures the workers' skills and put them into classes only once, it is labeled with "Unknown Immutable." If a method regularly checks the skills and changes the memberships accordingly, it is labeled with "Mutable." An example with task classes is as follows: If a complex workflow is required in an application, the workflow may contain different kinds of tasks, such as find, fix and verification tasks [3], which are connected to each other in the workflow. If the solution takes as input such a workflow and does not decompose it, it is labeled with "known Immutable." If a method decomposes the tasks into smaller ones, it is labeled with "Mutable." Another example is the case where we have a set of data labeling tasks and find a subset of the data labeling tasks appropriate for training AI workers. Then the set of tasks will be a new task class. A method to find such a subset is either "Unknown Immutable" or "Mutable" depending on whether it updates it during the execution or not.

B. Class-level Dynamicity of Task and Worker Classes. The dimensions represent the interaction between task and worker classes with the three categories:

Fixed The solution assumes that the set of task or worker classes is fixed and always contains their instances.

Independent The solution allows that task (or worker) classes can be added or deleted during the execution, and such operation is done independent of worker (or task) classes.

Responsive The solution adds or deletes tasks or worker classes during the execution, and the action is affected by the situation in their counterpart (e.g., changes in worker classes cause the division of task classes).

For example, a task decomposition method may not look at available workers at all. Such a method is labeled with "Independent." Another may consider the current availability of workers before the task decomposition.

Such a method is labeled with "Responsive." Responsiveness is definitely the potential benefit of computational division of labor, which will be useful in situations such as COVID-19 pandemic where we encounter a sudden change in labor resources.

Figure 4 puts some of the related topics (details will be shown in Figure 6) in the three dimension space. The results show that, at this moment, there are a limited number of studies that deal with responsive class generation. They deal with the two cases where (1) they assume human workers only and do not deal with automatic task decomposition and workflow optimization, and (2) they deal with human and AI workers for simple data labeling tasks. In contrast, there are few studies that deal with dynamic task and worker classes containing AI workers. Many studies focus on dynamicity of either worker or task classes only. Among the three benefits of division of labor, few studies focus on the problem of lowering switching cost. Some address the problem of improving workers' skills, and having Machine (AI) take place of manual labor, but the they addressed the problems independently. Most of the objective functions are defined in terms of short-term, requester-centric views on each workflow.

C. Controller of Division of Labor. In addition, there are different approaches on who controls the division of labor. The followings are potential subjects that control the division of labor process.

Boss There are approaches that assume a subject other than workers, who mainly takes care of the task decomposition, assignment, recruitment, etc. The boss can be a human, an AI agent, or a human-in-the-loop algorithm.

Colleague There are some cases where one of the workers decides who perform what tasks. For example, we can view active learning methods in the division of labor perspective as follows. We have two types of workers (a machine learner and humans), where the decision maker who assigns tasks for data labeling is the learner (Figure 5).

No one There is nobody who explicitly controls the division of labor. Rather, how it goes is incorporated in the design of the framework, such as incentive design, to implement "invisible hands." The task decomposition, assignment, education, recruitment, etc. are implemented as the result of every participant's action in the process of pursuit of their own gain.

3.2 Computational Division of labor View of Existing Studies

Figure 6 shows some of the related works that satisfy one of the following conditions: (1) The work deals with more than one task or worker classes (2) the work addresses issues related to benefits of division of labor - increasing workers' capabilities, lower switching cost, and AI workers taking place of manual labor.

Quality-Aware Microtask Assignment. For finite pool data categorization, there are approaches to assign appropriate tasks to different classes of workers to improve the result quality. For example, [18] discusses how to assign the categorization tasks to two categories of human workers, namely, experts and crowd workers, and an AI worker (a classifier model), in order to achieve high quality results. In the method, the AI worker is considered as a worker that responds to all the tasks that have been assigned to no human workers when the budget is run out. The method dynamically estimates the result quality in a situation that we train the AI worker with tasks labeled by humans and assign the rest of tasks to the trained AI worker. In terms of class-level dynamicity, it does not change task or worker classes during the execution. Furthermore, it always assumes to use a particular AI worker whose algorithm is known and does not accept AI workers that are developed by crowd workers.

Spatial Crowdsourcing. In spatial crowdsourcing, tasks and workers are associated to locations. Logically, we can think of many subclasses of each of the task and worker class, defined by their locations. The (sub-)class of each task or worker is known because we know their locations. In some settings, workers move thus the the

Topic	Examples	Task Classes and Relationship with their Instances	Worker Classes and Relationship with their Instances	Class-level Dynamicity and Interaction	Objective Functions and Constraints	Controller
Quality-Aware Microtask Assignment	[18]	Three subclasses (Crowd, Expert, AI) of a data labeling task class. Mutable	Three classes (Crowd, Expert, AI). Known Immutable	No change at task and worker class levels.	Better quality with a limited budget	Boss (AI)
Spatial Crowdsoucring Task Assignment	[29]	Subclasses of a spatial task class with different locations. Known Immutable	Classes for human workers in different locations. Known Immutable or Mutable	Task and Worker classes can be added and deleted dynamically, but independent of each other.	Maximizing total number of assignments, Optimized for average performance, Maximizing total payoff, etc.	Boss (AI)
Active Learning	[31, 22, 32, 9]	Subclasses (easy and hard) of a data labeling task. Mutable	Two classes (AI and Human) known Immutable.	No change at task and worker class levels.	Better machine learning models	Colleague (AI)
Working with a Large Number of Data Labeling AI Workers	[13]	Subclasses of a data labeling task class for human and AI workers. Mutable	One human worker class and many AI worker classes for different skills. Unknown Immutable	Automatic subclass generation and assignment triggered by dynamic estimation of AI worker performance	Quality and speed of task completion	Boss (AI)
	[11]	Two classes for easy and hard labeling tasks. Unknown Immutable.	Two classes for AI and Human. Known Immutable	Task classes are computed once based on a task prioritization algorithm before assignment	Better performance of AI Workers	Boss (AI)
Task Decomposition	[14]	Many dynamically-generated classes. Mutable.	One (human workers). Known immutable.	Task classes are dynamically generated, according to workers' judgement.	Completing tasks with the crowd. Each task must be done for a fixed price	Boss (human workers)
Team Formation	[30]	Many classes with different roles. Anyone can propose new role and edit role structure. Mutable.	Two (Team leader and team members). Known Immutable	Task classes are first defined by the leader, but can be reconfigured manually by anyone during the execution.	Organizing the teams to accomplish complex work with deadline of six weeks and budget.	Colleague (Human) and Boss (Human)
	[21]	Complex workflow consists of several tasks that require different capabilities. Known Immutable	Many classes with different capabilities and wage expectation. Known Immutable	No change at task and worker class levels. The algorithm optimally forms groups with available workers pool	Worker-worker affinity and upper critical mass with skill and cost constraint	Boss (AI)
Human-Factor Aware Microtask Assignment	[15]	One for speech transcription. Known Immutable	N subclasses. Workers are divided into each subclass based on current skill distribution. Mutable	No change at task and worker class levels.	Skill, Psychological Stress	Boss (AI)
	[28]	Each task will be broken down by the mentor into several subtasks only once. Unknown Immutable.	Two (Mentor and Intern worker). Known Immutable.	No change at task and worker class level.	Better learning of intern worker	Boss (Human)
	[20]	Many: each task has different content and requirements associated to keywords. Known Immutable	Many: each worker has different interests associated to keywords. Known Immutable	No change at task and worker class level. However, correspondences between task and worker classes are dynamically changed.	Worker Motivation, Task relevance, and Task diversity	Boss (AI)
Educational Process between Human and AI	[24, 17, 5, 10]	Two classes (tasks for education and not) of data labeling tasks. They are Known Immutable in [10], and Mutable in the others.	Two classes (Human and teaching AI). Known Immutable.	No change at task and worker class level.	AI learns effective teaching schemes and humans get higher ability to the task.	Colleague (AI)
Crowd Databases	[6, 19]	Many Tasks (filtering, join, etc.). Mutable	Two (Humans and DBMS). Known Immutable	Task classes are determined in the optimization phase based on the data statistics and fixed before the execution.	Monetary cost, Quality	Colleague (AI)

Figure 6: Division-of-Labor view of some of related research topics that deal with at least more than one worker or task class or address issues related to benefits of division of labor

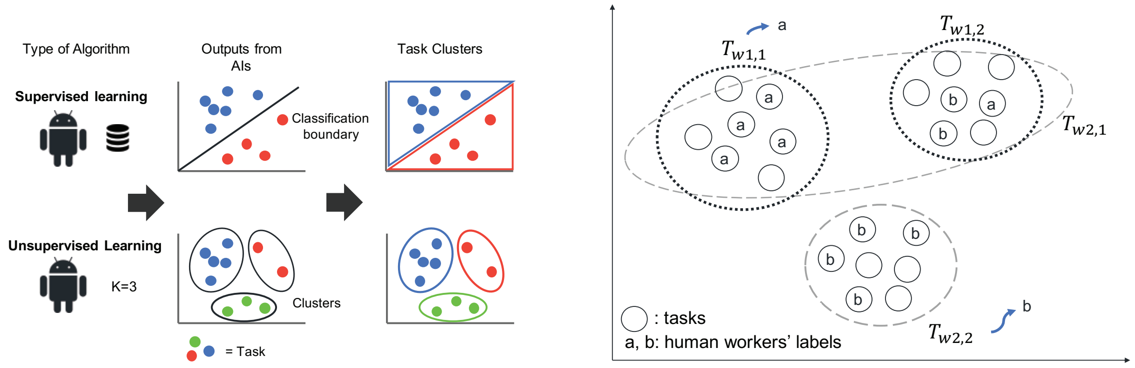


Figure 7: The method proposed in [13]. Each AI worker is assumed to cluster tasks, regardless of how it is implemented (left). The method conducts statistical tests to find whether all tasks in each task cluster is associated to a particular label, by looking at the labels given by human workers (Right). Here, $T_{w_{i,j}}$ means the jth task cluster of AI worker i. a and b are labels given by human workers.

worker classes they belong to are mutable. The task (worker) classes can be added and deleted dynamically as their instances appear and disappear, independently of workers (tasks).

Active Learning. Active learning interacts with human workers to (i) ask humans to make training data for data instances specified by *an AI worker* and (ii) receive the training data from humans to better learn the way to work [26]. Some active learning methods are aware of the performance of each worker and assign tasks to specific workers [22, 31]. In this discussion, we need to clearly distinguish the two roles of AI workers; a controller of task allocation to task classes and a task executor. The task allocation controller is a subject that organizes the dynamic allocation of data labeling tasks to the task class for human workers (i.e., hard task class), while a task executor is a subject that accomplishes data labeling tasks that could be assigned to human workers. From the viewpoint of the division of labor between humans and AIs, it is helpful to recognize that these active learning systems play two different roles because we potentially can find more flexible way to design AI components, e.g., working with a large number of task executor AIs, which we explain later. While the typical goal of active learning is to take over the human labor on data labeling tasks as a task executor, the allocation function is elaborated in some active learning models proposed so far. For example, Fang et al. [5] proposed an active learning model that encourages human learning by selecting a pair of workers having different skills to work together on the same data labeling task. Another example can be found in active learning that dynamically estimates the difficulty of tasks and assign only difficult tasks to workers of domain expert for minimizing the labor cost [9, 18, 32]. In terms of class-level dynamicity, no change happens on the task an dworker classes in active learning systems.

Working with a Large Number of Data Labeling AI Workers. There are studies on labeling a finite number of data items with not only human workers but also *a large number of* AI workers. The paper [13] proposes HACTAP (Human+AI Crowd Task Assignment Problem) that allows black box AI workers to join the workflow during its execution and assigns tasks to them if the high quality result is expected. Figure 7 shows the proposed method shown in [13]. Their method does not assume any particular model implemented in each AI worker. Rather, they assume that each AI worker outputs task clusters (which correspond to task (sub) classes in this paper) (Figure 7(left)). A task cluster will be meaningful if all tasks in each task cluster are associated to the same label. Therefore, the method conducts statistical tests to know whether each cluster corresponds to a particular label, by looking at the labels given by human workers. If the task cluster passes the test, all tasks in the cluster will be assigned to the AI worker. The method allows dynamic task assignment according to the available AI workers at each time, but the workflow is limited to a simple labeling task. The paper [11] presents a batch prioritization of data labeling tasks that allows a large number of black box AI workers to be efficiently trained. It statically assigns tasks to humans in advance before the task execution so that it effectively train AI workers independent of their underlying models. Thus, it can train a large number of blackbox AI workers with different underlying models in parallel. In terms of class-level dynamicity, there are only two fixed classes for tasks and workers. However, their hard labeling tasks are carefully chosen so that the tasks are effective for training AI workers with any underlying models.

Task Decomposition. There are studies to ask human workers to do task decomposition. For example, [14] proposes the PDS (Price-Divide-Solve) algorithm to ask crowd workers to decomposed tasks into smaller ones. The result workflow contains a diverse set of microtasks whose results are merged to produce the final product. Worker classes are not explicitly dealt with in the algorithm, but they assume that each task is done with the fixed payment (20 cents, in their implementation). This implies that it assumes that the task is easy enough so that we can easily find workers that do the task with the payment in crowdsourcing platforms.

Team Formation. There are studies on how to configure teams to solve complex problems. [30] proposed flash-organization which enables us to hire expert crowd workers into role structures and dynamically reconfigure the structure via version control. [21] proposed an optimization model for task assignment in a collaborative crowdsourcing environment and proposed optimization algorithms with theoretical guarantees. From the aspect of the computational division of labor, [30] allows human workers to change task classes, but it is not automatic, while [21] performs automatic task assignment at class-level, but does not allow dynamic change at the class level. Assignment to AI workers is not discussed in both of them.

Human-Factor Aware Microtask Assignment. While many crowdsourcing studies have assumed that anonymous workers have the same role, every human being is different, i.e. what tasks the workers are good at, what motivates them, and what they are doing the task for, are different for each person. Thus, considering the human rights of workers, it is natural to take into account the worker's perspective, in other words, it is important to consider who, when, and which task should be performed by whom. Therefore, the number of worker class is usually plural in this topic. Recently, the importance of human-factor in crowdsourcing has been argued [2], and task assignment research has been addressed in line with this perspective. For example, [36] considers psychological stress, [20] considers motivation, and [28] considers worker's learning. Class-level dynamicity is not addressed in this topic.

Educational Process between Humans and AI. Assigning a task to workers with appropriate ability is important for efficient problem solving. Increasing workers' skills is an important issue in division of labor, from the viewpoint of obtaining high-quality task results in the long term. For this problem, there are studies dealing with the interactive educational process between humans and AI. In this process, the AI learns the optimal task assignment to maximize the learning effect for human workers. These researches give us new insight into the task division from the viewpoint of the cultivation of expert human workers. However, the existing studies discuss improving skills *within* a particulars task class. From the division of labor view, it is important to address the problem of improving skills across task classes in different workflows.

Crowd Databases. Crowd Databases such as CrowdDB[6] and Deco[19] employ human workers to obtain data that are not stored in the current snapshot of database in the storage. The workflow contains a variety of tasks such as data entry, selections, join, ordering, while it does not explicitly deal with worker classes and attributes. The optimization of workflow is based on the data statistics, rather than worker availability and their skills.

4 Challenges and Open Problems

As shown in Section 3, although there have been many studies that address topics relevant to computational division of labor, this area is still in its infancy. In most studies, problems such as task decomposition, worker recruitment and education, are discussed with particular assumptions on available workers and decomposed tasks. There are only a few studies that deal with dynamic interactions between the skills of available workers and task decomposition, and that focus on benefits of division of labor in their objective functions. Given the current status, this section discusses challenges and some of open problems in computational division of labor with human and

AI workers.

4.1 Workers to Tasks

Knowing Relevant and Qualified AI Workers. The decomposed tasks include those to be performed by human and AI workers. However, AI workers are more diverse to each other in the skills than human workers. If a task is given, finding AI workers that we can be employed in its decomposed tasks will be a challenge.
Worker-Conscious Task Decomposition. In the existing research, the task decomposition is conducted once before the execution, assuming the simple assumption on workers (e.g., there are enough number of workers in the worker pool who are able to perform every task). Task decomposition schemes that are more conscious of the available workers in the pool will be an interesting issue. Another interesting issues is the tailor-made task extraction schemes; if a worker shows an interested in the project, the system extracts a task for him considering her skills and other constraints. If we include AI workers in the worker pool, workers will be more diverse in their speed, skills, and appropriate interfaces. An AI worker handles a bundle of tasks better than performing each task one by one. We need to deal with such a diversity.
On-the-Fly Workflow Switch without Stopping Its Execution. The situation of worker pool sometimes changes as time goes. For example, workers in Japan usually sleep at night in Japan Time and there will be a lower number of workers who can process Japanese. If a pandemic happens, worker distributions will dramatically change. Reassembling tasks and switch to new ones should be done without stopping its execution, while keeping a certain service level. When workers change, it would be inefficient to calculate the optimal solution from scratch each time a change occurs. Therefore, optimization mechanism that adapts to changes of workers and tasks, e.g., incremental algorithm, is becoming more important than before. As a preceding study, an incremental algorithm for finding an optimum worker assignment when a worker set changes is proposed [23].

4.2 Tasks to Workers

Incentive Design for Recruiting and Developing AI Workers. Tasks registered in the task pool for human workers do not necessarily require human workers and sometimes can be processed by AI workers. For example, active learners that are appropriately trained with crowdsourced labels sometimes output good quality results [31]. In some cases, we may be able to ask AI workers to performs most of a tremendous number of tasks. If we gave a good incentive to people, they would search for or develop their AI workers to perform the available tasks. The open question is how to design such an incentive. Effective and fair payment framework for AI developers needs to be investigated.
Psychological Stress Management of Human Workers. Job change is one of the things that give psychological stress to people [7]. Therefore, the responsiveness introduced by computational division of labor may cause additional psychological stress. We need to take into consideration the skill of workers, the types of tasks they have done so far, and their long-term career plans.
Matching of Supply and Demand of Skills. Mismatch of supply and demand of skills cause problems in lack and excess in labor resources. Developing ways to make the demand of the required skills visible will affect workers on choosing skills to learn and designing their long-term career for their future.
Education Strategies for Human and AI Workers. Most existing research for educating people in crowdsourcing settings all targets a particular set of microtasks. Extracting common skills from the task markets and provide educations for workers will be indispensable for the efficient learning and education strategies.

4.3 Holistic Perspective

Integration of Human and AI Worker Results. We naturally obtain diversity with human workers. Therefore, many existing studies on integrating results from crowd workers assumes the diversity. In contrast, a set of AI workers may implement similar algorithm and we may not able to consider them to make completely independent decisions.

Human-in-the-Loop Division of Labor Algorithms. Division of labor itself can be implemented with AI and human computation. Many problems in computational division of labor themselves can be regarded as "tasks." Thus, they can be implemented with AI and human computation. For example, "optimization algorithm for task decomposition" in which workers contribute their computational power to some part of optimization would be an algorithm design of great interest.

Social-Level Objective Functions. Most of related literature that address optimization problems has objective functions at a requester or a worker level. However, optimization focusing on the project-level efficiency only often concludes that working with only a few high performers is the best solution. In addition, platform-based recruitment of workers often cause price discrimination and exclusion of particular groups of workers [25]. Taking care of social-level objective functions, such as achieving inclusive labor markets, will be an important open problem.

5 Conclusion

In the recent years, more and more information on tasks is being circulated on the internet, and there are many labor resources accessible through it. In addition, we see the rapid growth of AI technology that allows us to have them workers for our tasks. This motivates us to take the computational approach to division of labor, in which we use AI agents that implement algorithms to assign appropriate tasks to human and AI workers to achieve specific goals. This paper investigated problems on computational division of labor around crowdsourcing and data-centric human-in-the-loop systems research. We explained what computational division of labor problems deal with and introduced a set of dimensions and terms to classify existing solutions for problems related to this topic. We identified the current status of this topic and showed that there are a number of interesting research challenges.

Acknowledgment

This work was partially supported by JST CREST Grant Number JPMJCR16E3, Japan.

References

[1] S. Amer-Yahia et al. Making ai machines work for humans in fow. *ACM SIGMOD RECORD (to appear)*, 2020.

[2] S. Amer-Yahia and S. B. Roy. Toward worker-centric crowdsourcing. *IEEE Data Eng. Bull.*, 39(4):3–13, 2016.

[3] M. S. Bernstein, G. Little, R. C. Miller, B. Hartmann, M. S. Ackerman, D. R. Karger, D. Crowell, and K. Panovich. Soylent: a word processor with a crowd inside. In *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology, New York, NY, USA, October 3-6, 2010*, pages 313–322, 2010.

[4] V. Dhar. When to trust robots with decisions, and when not to. *Harvard Business Review*, May 2016.

[5] M. Fang, X. Zhu, B. Li, W. Ding, and X. Wu. Self-Taught active learning from crowds. In *2012 IEEE 12th International Conference on Data Mining*, pages 858–863, Dec. 2012.

[6] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. Crowddb: answering queries with crowdsourcing. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2011, Athens, Greece, June 12-16, 2011*, pages 61–72, 2011.

[7] J. GREUBEL and G. KECKLUND. The impact of organizational changes on work stress, sleep, recovery and health. *Industrial Health*, advpub:1102240056–1102240056, 2011.

[8] D. Gross-Amblard, A. Morishima, S. Thirumuruganathan, M. Tommasi, and K. Yoshida. Platform design for crowdsourcing and future of work. *IEEE Data Eng. Bull.*, 42(4):35–45, 2019.

[9] S. Hao, S. C. H. Hoi, C. Miao, and P. Zhao. Active crowdsourcing for annotation. In *2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, pages 1–8. IEEE, Dec. 2015.

[10] E. Johns, O. Mac Aodha, and G. J. Brostow. Becoming the expert - interactive multi-class machine teaching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2616–2624, 2015.

[11] M. Kimura, K. Wakabayashi, and A. Morishima. Batch prioritization of data labeling tasks for training classifiers. In *Proceedings of the Eighth AAAI Conference on Human Computation and Crowdsourcing,HCOMP 2020*, 2020.

[12] M. Kobayashi, H. Morita, M. Matsubara, N. Shimizu, and A. Morishima. An empirical study on short- and long-term effects of self-correction in crowdsourced microtasks. In *Proceedings of the Sixth AAAI Conference on Human Computation and Crowdsourcing, HCOMP 2018, Zürich, Switzerland, July 5-8, 2018*, pages 79–87, 2018.

[13] M. Kobayashi, K. Wakabayashi, and A. Morishima. Quality-aware dynamic task assignment in human+ai crowd. In A. E. F. Seghrouchni, G. Sukthankar, T. Liu, and M. van Steen, editors, *Companion of The 2020 Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 118–119. ACM / IW3C2, 2020.

[14] A. Kulkarni, M. Can, and B. Hartmann. Collaboratively crowdsourcing workflows with turkomatic. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, CSCW '12, pages 1003–1012, New York, NY, USA, 2012. ACM.

[15] K. Kumai, M. Matsubara, Y. Shiraishi, D. Wakatsuki, J. Zhang, T. Shionome, H. Kitagawa, and A. Morishima. Skill-and-stress-aware assignment of crowd-worker groups to task streams. In *Proceedings of the Sixth AAAI Conference on Human Computation and Crowdsourcing, HCOMP 2018, Zürich, Switzerland, July 5-8, 2018.*, pages 88–97, 2018.

[16] R. Kurzweil. The law of accelerating returns, 2001. kurzweilai.net/the-law-of-accelerating-returns.

[17] W. Liu, B. Dai, A. Humayun, C. Tay, C. Yu, L. B. Smith, J. M. Rehg, and L. Song. Iterative machine teaching. *arXiv preprint arXiv:1705.10470*, 2017.

[18] A. T. Nguyen, B. C. Wallace, and M. Lease. Combining crowd and expert labels using decision theoretic active learning. In E. Gerber and P. Ipeirotis, editors, *Proceedings of the Third AAAI Conference on Human Computation and Crowdsourcing, HCOMP 2015, November 8-11, 2015, San Diego, California, USA*, pages 120–129. AAAI Press, 2015.

[19] A. G. Parameswaran, H. Park, H. Garcia-Molina, N. Polyzotis, and J. Widom. Deco: declarative crowdsourcing. In *21st ACM International Conference on Information and Knowledge Management, CIKM'12, Maui, HI, USA, October 29 - November 02, 2012*, pages 1203–1212, 2012.

[20] J. Pilourdault, S. Amer-Yahia, S. B. Roy, and D. Lee. Task relevance and diversity as worker motivation in crowdsourcing. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pages 365–376. IEEE, 2018.

[21] H. Rahman, S. B. Roy, S. Thirumuruganathan, S. Amer-Yahia, and G. Das. Optimized group formation for solving collaborative tasks. *The VLDB Journal*, 28(1):1–23, 2019.

[22] F. Rodrigues, F. Pereira, and B. Ribeiro. Gaussian process classification and active learning with multiple annotators. 32(2):433–441, 2014.

[23] S. B. Roy, I. Lykourentzou, S. Thirumuruganathan, S. Amer-Yahia, and G. Das. Task assignment optimization in knowledge-intensive crowdsourcing. *VLDB J.*, 24(4):467–491, 2015.

[24] C. P. G. Runzhe Yang, Yexiang Xue. Pedagogical value-aligned crowdsourcing: Inspiring the wisdom of crowds via interactive teaching. *International Conference on Autonomous Agents and Multiagent Systems(AAMAS)*, 2018.

[25] S. S. Sara C. Kingsley, Mary L. Gray. Monopsony and the crowd: Labor for lemons? In *Proceedings of the Internet, Policy & Politics Conference (IPP2014)*, 2014.

[26] B. Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

[27] A. Smith. *An Inquiry into the Nature and Causes of the Wealth of Nations*. McMaster University Archive for the History of Economic Thought, 1776.

[28] R. Suzuki, N. Salehi, M. S. Lam, J. C. Marroquin, and M. S. Bernstein. Atelier: Repurposing expert crowdsourcing tasks as micro-internships. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 2645–2656, 2016.

[29] Y. Tong, Z. Zhou, Y. Zeng, L. Chen, and C. Shahabi. Spatial crowdsourcing: a survey. *VLDB J.*, 29(1):217–250, 2020.

[30] M. A. Valentine, D. Retelny, A. To, N. Rahmati, T. Doshi, and M. S. Bernstein. Flash organizations: Crowdsourcing complex work by structuring crowds as organizations. In *Proceedings of the 2017 CHI conference on human factors in computing systems*, pages 3523–3537, 2017.

[31] Y. Yan, R. Rosales, G. Fung, and J. G. Dy. Active learning from crowds. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 1161–1168, 2011.

[32] C. Zhang and K. Chaudhuri. Active learning from weak and strong labelers. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 703–711. Curran Associates, Inc., 2015.

Federated Learning in the Lens of Crowdsourcing

Yongxin Tong, Yansheng Wang, Dingyuan Shi
SKLSDE Lab, BDBC and IRI, Beihang University, China
{yxtong, arthur_wang, chnsdy}@buaa.edu.cn

Abstract

The success of artificial intelligence (AI) is inseparable from large-scale and high-quality data, which is not always available. Involving human forces like crowdsourcing can help provide more training data and improve data quality for AI tasks. But with more privacy concerns and stricter laws, the data isolation problem is becoming worse, just when federated learning (FL) has emerged as a promising solution. In this article, we highlight the core issues in federated learning in the lens of crowdsourcing, including privacy and security, incentive mechanism, communication optimization and quality control. We expect to inspire the design of federated learning systems with existing crowdsourcing techniques. We also discuss emerging future challenges to implement a fully fledged federated learning platform.

1 Introduction

Artificial intelligence (AI) has come to a golden age. With the help of big data, new learning algorithms and powerful computing hardware, AI has shown huge potential in many real-life applications, such as image recognition and text processing. However, its success highly relies on large-scale and high-quality training data, which is not always available.

Involving human forces proves effective in either providing more training data or improving the data quality for AI tasks. In particular, crowdsourcing [1, 2], is one of the most practical solutions to data problems in AI. It is a computation paradigm where humans are gathered to collaboratively accomplish easy tasks. A representative example of crowdsourcing empowered AI is the famous ImageNet project [3], where most pictures are labeled by crowdsourced workers. The Amazon Mechanical Turk (AMT) is one of the most successful commercial crowdsourcing platforms, where a large number of data labeling tasks with monetary rewards are provided by AI practitioners for freelancers.

The lack of large-scale training data is becoming more severe in recent years. In many industries, data are often isolated by different companies or organizations. Because of commercial competition and administrative issues, they would not like to share their data. They have to train models separately with their own data but the performance is often unsatisfactory due to the lack of data. Meanwhile, with people's increasing awareness on data security and individual privacy, data privacy in AI is becoming increasingly important. Many countries are enacting strict laws to protect the data privacy of their citizens. For example, EU's General Data Protection Regulation (GDPR) which was enforced on May 25, 2018, has stipulated that any use of personal data in a company must be authorized by the data owners. Therefore, privacy issues exacerbate the data isolation problem.

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

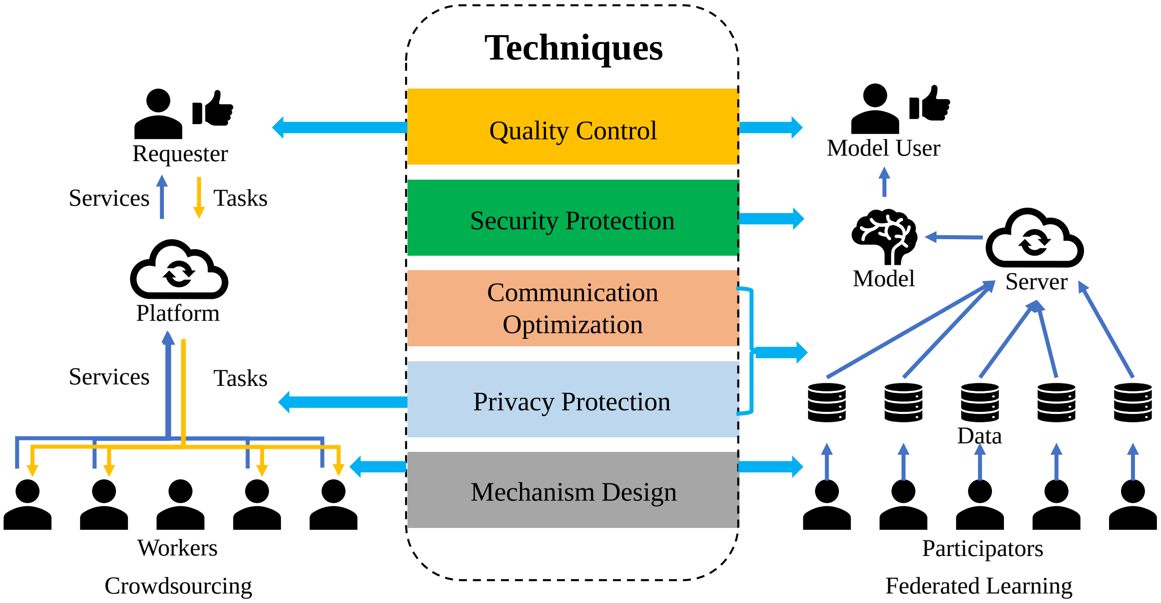


Figure 1: Comparison between crowdsourcing and federated learning.

Federated learning (FL) [4, 5, 6] has emerged as a promising solution to the data isolation problem in AI. First proposed by Google, FL aims to collaboratively build machine learning models with data from massive mobile devices without violating data privacy. As with crowdsourcing, FL also organizes humans, their devices and data, to collaborate on specific tasks. Therefore, FL can be considered as a special form of crowdsourcing. In this paper, we propose to understand federated learning from the lens of crowdsourcing. The characteristics of FL in the core issues of generic crowdsourcing and the unique issues of FL are summarized as below.

- **Privacy and security.** The most important issue in FL is privacy protection, whereas privacy is of less concerns in general crowdsourcing. Secure machine learning algorithms often play a central role in FL.
- **Incentive mechanism.** Both FL and general crowdsourcing need to motivate participants. However, there are two differences. The incentive mechanism in many crowdsourcing applications is based on the Business-to-customer (B2C) mode while in FL it can also be the Business-to-business (B2B) mode, as the participators in FL can be different companies. The incentive procedure in crowdsourcing is often a single round, while in FL it takes multiple rounds following the training steps in machine learning, which makes the design of incentive mechanisms more difficult.
- **Communication optimization.** In crowdsourcing the communication overhead is usually not a problem as each participant only has to submit small data (like the label of a picture) for a single round. However, in FL it can be a core issue because the training process often involves frequent communication of high dimensional gradients.
- **Quality control.** Optimizing the collaboration results is crucial both in crowdsourcing and FL. The difference is that in crowdsourcing the task is simple and the focus is to improve the accuracy of integrated results with a constrained budget. The task in FL is more complicated, and the focus lies in how to deal

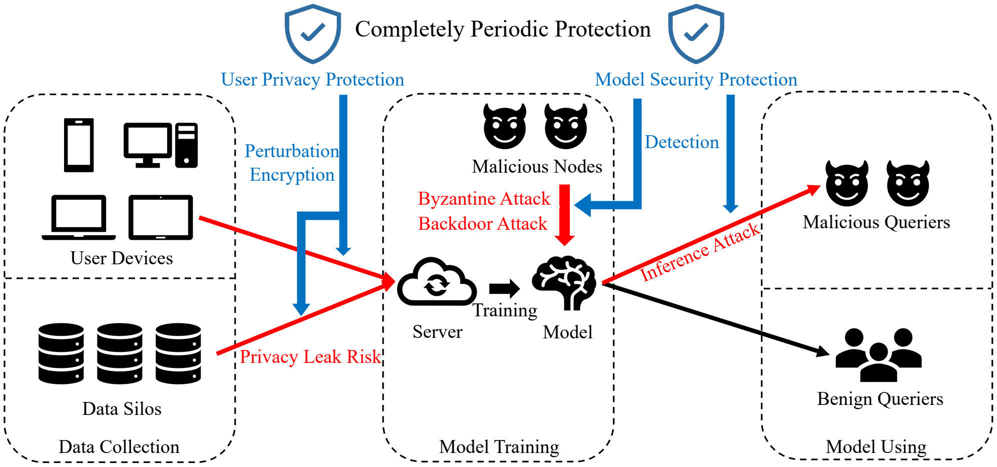


Figure 2: Completely periodic protection in federated learning.

with the heterogeneity of different parties such as non independently and identically distributed data and imbalanced computation resources.

- **Task assignment.** Task assignment is a core component in a crowdsourcing platform [7, 8], which needs to manage massive tasks and workers and to effectively allocate the resources. The assignment results can decide the practical performance of the platform. In FL, task assignment may not be an issue, as there is only one learning task in most cases.

This paper discusses the above core issues in federated learning from the perspective of crowdsourcing. The aim is to inspire the design of federated learning systems with existing techniques in crowdsourcing platforms. We also pinpoint future challenges to implement a fully fledged federated learning platform following the principles of crowdsourcing.

2 Privacy and Security

In general crowdsourcing, workers and users only provide some necessary information such as worker skills or positions which only leaks little privacy. Existing privacy protection techniques like anonymization and encryption are sufficient to protect such information. However, in federated learning, the protection object becomes massive user data which is more sensitive and easier to leak privacy when external knowledge is used. Furthermore, compared with crowdsourcing, federated learning makes it harder to judge the benignity of user uploads. This is because machine learning models are black boxes and it is non-trivial to explain the contribution of user uploads. Accordingly, it is possible for malicious users to upload information and thwart model training.

A federated learning system is expected to offer periodic protection on both user data privacy and model security, as shown in Figure 2. Specifically, a safe-to-use federated learning framework should *(i)* collect and use user data privately and *(ii)* ensure that the model converges without poisoning and will not be stolen.

2.1 Data Privacy Protection

In federated learning, for each participator, the server and other participators cannot be easily trusted. Direct uploading of raw data can lead to privacy leaks. Uploading model parameters instead of raw data seems safer, which is also allowed in the most recognized algorithm FedAvg [5]. Some recent findings, however, show that only with model parameters, a malicious attacker can still deploy inference attack to judge membership or even reconstruct the raw data [9]. Therefore, we need to design privacy protection protocols for federated learning, which mainly include two techniques: perturbation and encryption.

Perturbation. Perturbation techniques require participators to inject noise to their raw data or intermediate results, making others hard to infer what they have uploaded. To quantitatively analyze the degree of noise injection, a widely accepted measure is differential privacy (DP), proposed by Dwork [10]. Its idea is to quantify the degree of indistinguishability by probability distributions. The DP measurement is firstly brought up and applied in database. In the field of federated learning, however, a big challenge derives from the long iterative process and from massively distributed data. For each participator and in each iteration, DP should always be satisfied. Ensuring such strict privacy protection requires strong noise to be injected, which severely deteriorates data accuracy. To prevent privacy cost from boosting wildly with iteration rounds, moments accountant [11] has been proposed. It can make the privacy budget increase sub-linearly (square root) to iteration rounds. As for large node numbers, shuffle model [12] techniques effectively cut down privacy budgets by ensuring anonymity. Besides supervised learning, DP has also been applied to unsupervised algorithms like topic modeling [13, 14].

Encryption. Perturbation technique can be considered as a balance between privacy levels and data accuracy. Nevertheless, perfect secrecy cannot be achieved as exposure of little private information always exists. Encryption techniques, on the other hand, aim to directly circumvent such exposure, *i.e.*, to calculate via ciphertexts. Several existing encryption techniques can be applied to federated learning, such as secure multiparty computation (SMC), homomorphic encryption (HE) and garbled circuits (GC). Bonawitz et al use pseudo random vectors as a mask to cover the raw updates and those random vectors can neutralize each other after aggregation [15]. Using HE to protect user privacy demands a protocol via which the server can aggregate the ciphertexts from each device and finally decipher the result [16]. Besides software level solutions, hardware solutions are also worth considering such as the trusted execution environment (TEE).

2.2 Model Security Protection

Model security refers to that the model converges to a global optimum and the results can be used safely. Model security can be violated by multiple types of attacks.

Byzantine Attack. Byzantine attack is a classical attack in distributed networks. Attackers aim to disturb the model training process and make the model unable to converge. In FL, the malicious nodes (Byzantine nodes) will upload random vectors to mislead the aggregated gradient descent direction and thus obstruct the model convergence. The core idea to detect Byzantine attack is to evaluate and to spot the outliers among user uploads. Intuitively, examining the angle between different vectors' directions may be a solution [17]. Another intuitive way is to use the distance to median values as an outlier judgement and its effectiveness is also verified in [18]. By maintaining a non-Byzantine node set while training, the later solution reaches lower time complexity.

Backdoor Attack. By deploying Byzantine attack detection techniques, we can ensure the convergence of model training. However, some higher-level attackers (nodes) can cheat the Byzantine detector by uploading plausible updates and force the model to converge to a point where some subtasks or intentionally designed misclassification are achieved. For example, the attacker may hope to make a spelling prompt model always provide some specific words (A restaurant owner hopes her restaurant name be prompted after a user types "My favorite restaurant is..."). To realize this, unlike Byzantine attackers who simply upload random vectors, they will use data poisoning or model poisoning techniques. The purpose of data poisoning is to train models with

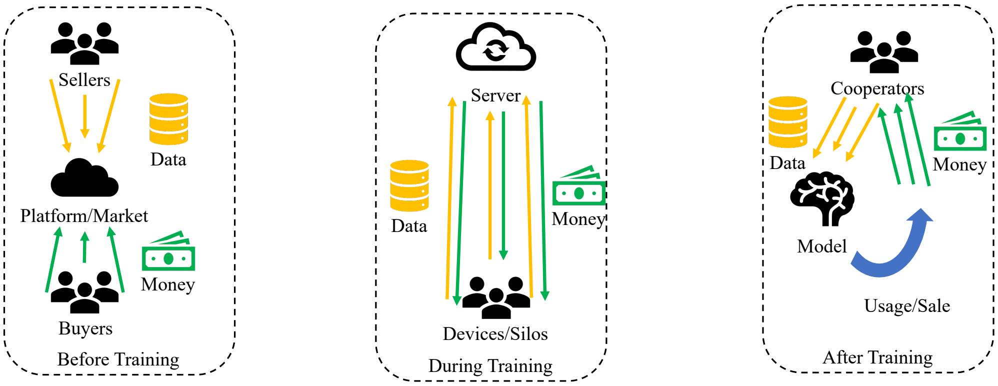


Figure 3: Mechanism design over all periods.

intentionally mislabeled or polluted data while model poisoning refers to that malicious nodes train and upload local models with training goals different from the global model. To tackle data or model poisoning, we can evaluate the approximated loss upper bound and remove outliers before local model aggregation [19].

Model Stealing. Equipped with Byzantine and backdoor attack detectors, a federated learning system is able to safely train a model. However, we also need to provide protection for model use. Model stealing happens when the model is confidential and can only be used via APIs (*i.e.*, Machine-Learning-as-a-Service, MLaaS). In that case, attackers may use the APIs to infer the model structure and parameters, and the confidential information of the model is exposed. In federated learning, this may happen when one of the participators wants to forcibly occupy all the outcomes. To prevent this, we can deploy a detector to examine the frequency of API queries and judge whether the query sequence is benign [20].

3 Incentive Mechanism

Both general crowdsourcing and FL involve multiple human participators. Thus suitable incentive mechanisms are necessary to attract people to actively contribute to the tasks. In traditional crowdsourcing, some tasks may be less attractive to workers due to distance, difficulty or other reasons. Therefore, the platform needs to motivate the workers with additional rewards [21]. For FL, incentive mechanism design is more difficult. This is because the black-box nature of many machine learning models makes it tricky to evaluate each participator's contribution. Furthermore, the number of participators in FL, especially in cross-device settings, can be much larger than in traditional crowdsourcing.

The mechanisms for FL should be incentive-compatible and fairness-aware. They can be accomplished before, during and after training process, as shown in Figure 3.

Mechanisms before Training. Designing mechanisms to incite participators before training means to establish suitable rules for data trading. In the past decade several data markets and data sharing platforms such as Dawex[1], Xignite[2] and WorldQuant[3] have been developed. They all hope to build a data trading platform where companies

[1] https://www.dawex.com

[2] https://www.xignite.com

[3] https://www.worldquant.com

and users can buy, sell and exchange data with satisfactory prices to all parties. However, this goal is hard to reach. As Fernandez et al point out, different combinations of data may produce different levels of values[22], so instead of setting static price for data, people seek to find dynamic evaluation methods.

Mechanisms during Training. The goal of designing mechanisms during training is to incite the participators to use their best data for training. To achieve the goal, participators who contribute the most deserve the highest reward. Many researchers model the training process as a Stackelberg game[23]. There are two stages in this setting. In the first stage, the server receives updates from each user and distributes rewards base on their contributions to the model. In the second stage, users change their update strategies based on the rewards they receive. Then the cycle repeats until all the server and users converge to equilibrium. Chen et al study mechanism design of federated learning from a game theoretical and optimization perspective. Recent years blockchain emerges as a novel technique and raises wide-spread research enthusiasm. It is also an alternative technique to design incentive mechanisms. One way to combine blockchain with mechanism design is to build a reputation record by the blockchain[24]. Owing to the immutability and consensus of blockchain, users' reputation would be hard to recover once it gets damaged and thus the participators will behave honestly to maintain their reputations.

Mechanisms after Training. In cross-silo federated learning settings the companies cooperate with each other to train a global model and the rewards (profits) mainly come from the model use. Distributing the profits fairly requires designing mechanisms that can evaluate each entity's contribution to the final model. Shapley value is a classical concept in game theory to evaluate contributions and can also be applied in profit sharing in federated learning. The drawback is that the calculation of Shapley values is time-consuming. As a result, Song et al propose a novel accleration technique to make it practical[25].

4 Communication Optimization

Crowdsourcing platforms usually do not care the communication overhead. Tasks like labeling the images require the workers to upload very few data for a single round. The workers can submit their results separately and occasionally. Hence there is not much pressure on the server's communication bandwidth. However, the communication cost becomes a primary bottleneck for FL. Model training algorithms like the stochastic gradient descent (SGD) take a large number of rounds to converge. Besides, unlike the powerful servers in distributed learning, the nodes in FL are massively distributed mobile devices with limited communication bandwidth and active time, which makes the problem even more challenging.

Many methods have been proposed to improve the communication efficiency of FL. Two basic ideas are to either reduce the number of interactions between parties and the server or to compress the data in transmission. These two ideas can also be combined. Some representative techniques are explained as below.

4.1 Interactive Number Reduction

There are three different ways to reduce the number of interactions during the federated learning process: client sampling, local updating, and decentralized training.

Client Sampling. In a cross-device federated learning scenario, there are usually millions of devices in participation, which makes round-robin strategies impossible to work. The server has to sample some parties in each round for faster convergence and the accuracy should be compromised. The client sampling trick usually does not work alone. In some earliest works of FL such as Federated Optimization [26] and FedAvg [5], random sampling over multiple parties is used during the training process. Afterwards, it is applied by other works on cross-device FL by default. Most of them still use the simplest uniformly random sampling method. Some consider choosing the clients conditionally. For example, FedCS [27] actively manages the clients based on their resource conditions. Reinforcement learning techniques have also been applied to adaptively sample more suitable clients [28].

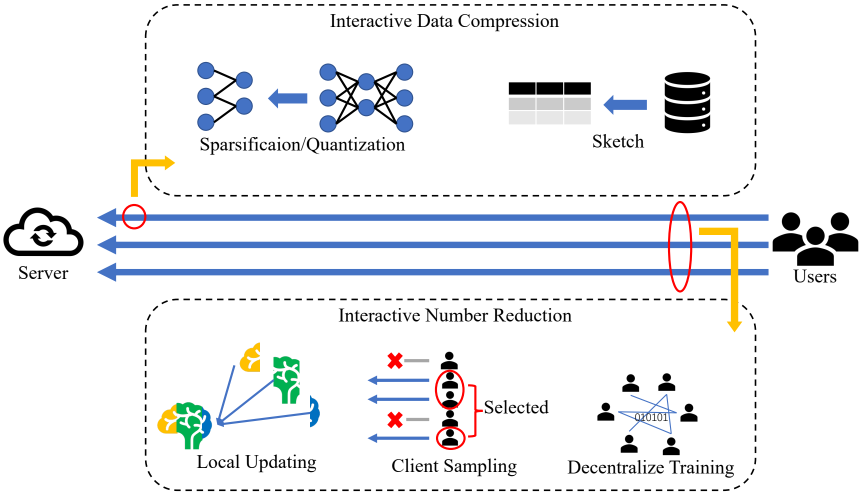

Interactive Data Compression
Sparsificaion/Quantization
Sketch
Server
Users
Interactive Number Reduction
Local Updating
Selected
Client Sampling
010101
Decentralize Training

Figure 4: Communication optimization techniques in federated learning.

Local Updating. The local updating techniques are first proposed to improve the communication efficiency in distributed learning and they can also work well in FL. Some approaches use primal-dual methods to decompose a global objective into many subproblems that can be solved in parallel. Therefore the communication rounds can be effectively reduced. However, in FL some trivial ideas have shown good empirical performance. The most commonly used algorithm FedAvg [5] is based on local SGD and global averaging. An extension [29] uses more adaptive local updating methods like ADAGRAD, ADAM and YOGI.

Decentralized Training. Decentralized learning algorithms can effectively reduce the communication overhead on the server side by apportioning them to each client. Many existing works on decentralized learning can also be applied in FL, such as decentralized training algorithms of linear classifiers [30] and deep neural networks [31]. Some other works are based on special network structures. In [32], it considers the problem that the distributed nodes can only communicate to their neighbors on a fixed communication graph in machine learning and devises a gossip algorithm for average consensus with convergence guarantees. In [33], a dynamic peer-to-peer network environment is considered and a novel decentralized FL framework BrainTorrent is proposed.

4.2 Interactive Data Compression

Another way to reduce communication cost is to compress the data in transmission directly. Many existing FL works follow the model compression approaches in machine learning, such as sparsificaion and quantization. Some others apply data structures like sketch to re-encode the gradients.

Compression with Sparsificaion and Quantization. One of the pioneering works in FL [4] focuses on improving the communication efficiency by random mask structures as well as a combination of quantization, random rotations, and subsampling to compress the model parameters. The distributed mean estimation problem is studied in [34] and an communication-efficient compressing algorithm using constant number of bits is devised. In [35], it proposes Deep Gradient Compression (DGC) in cross-device distributed learning setting which can greatly reduce the communication bandwidth by 99%. Two different strategies named extrapolation compression and difference compression are proposed in [36], combined with a decentralized training algorithm. A special quantization-based technique for gradient compression in FL is proposed in [37], combined with a periodic averaging learning algorithm.

Compression with Sketch. A probabilistic data structure for compressing big streaming data, the sketch [38], is also used in gradient compression. To find heavy hitters (most frequent items) in the federated setting, a sketch-based algorithm with local differential privacy is proposed in [39]. To compress sparse and nonuniform gradient in distributed machine learning, MinMaxSketch is designed in [40], which uses a quantile sketch to sort gradients into buckets. Another similar approach, the Sketched-SGD, is proposed in [41]. It is demonstrated to have a 40x reduction in total communication cost with no loss in final model performance. In [42], the authors prove that Count sketch without additional noise can satisfy the notion of differential privacy (DP) under specific assumptions, which is also known as "privacy for free".

5 Quality Control

The heterogeneity of workers in crowdsourcing results in the quality variation of aggregated results. The purpose of quality control is to quantify the heterogeneous quality of workers and tasks and effectively aggregate results to ensure high-quality task completion. Similarly, there is also the heterogeneity problem in FL participants, especially in cross-device scenarios. Thus quality control is also important for FL. Different from quality control techniques in crowdsourcing, which concentrate on evaluating the characteristics or skills of different workers to improve crowdsourced results, quality control in FL mainly deals with two unique challenges, the heterogeneity in data and the heterogeneity in resource.

5.1 Data Heterogeneity

The data heterogeneity mainly refers to the non-IID data problem in FL. In [43], a taxonomy of non-iid data regimes is provided, including feature distribution skew, label distribution skew, etc. The most commonly adopted FedAvg [5] algorithm only considers the IID case at first and no theoretical analysis on its convergence is made. To improve the performance of FedAvg especially in non-IID settings, some approaches have been proposed [44, 45, 46]. A more rigorous theoretical proof on the convergence of FedAvg with non-IID data is provided in [47]. It establishes a convergence rate by the inverse of the number of iterations for strongly convex and smooth problems. However, in [48], it demonstrates that the accuracy of FedAVG can be largely damaged on highly skewed non-IID data where each device only has a single class of data. The solution is to create a globally shared small subset of data. But if each device shares too much data, the privacy constraints might be broken, which is contradictory to the initial purpose of FL. To avoid data sharing, in [49], it proposes federated augmentation, where each participant will collectively train a generative model. Therefore, the local data can be augmented by the generative model and gets rid of the non-IIDness. The differences between each client's data distribution can also be taken as black boxes. A reinforcement learning-based client selection approach is proposed in [28] to deal with the non-IID data.

5.2 Resource Heterogeneity

The computation resources of different devices are commonly heterogeneous in mobile edge computing. The main objective is to optimize the model quality by resource allocation. The problems of resource heterogeneity such as uncertainty of wireless channels and devices with heterogeneous power constraints have been emphasized in [50]. It formalizes FL with heterogeneous resources as an optimization problem to capture the trade-offs between efficiency and accuracy. Then it solves the non-convex problem by decomposing it into several convex sub-problems. Control algorithms have also been proposed to realize more effective resource allocation. An adaptive control algorithm for FL with distributed SGD is proposed in [51]. It studies the convergence bound of the learning problem with the resource consumption constraints. A protocol named FedCS is proposed in [27] which performs client selection in each round according to their different resource conditions. Devices with poor computation power or low communication bandwidth will be eliminated during training. A more comprehensive survey on FL in mobile edge networks can be found in [52].

6 Future Directions

In this section, we will envision some future directions in federated learning.

Task Assignment. Until now, no work has considered the task assignment problem in FL. In most cases there is only one learning task. Meanwhile a single learning task cannot be decomposed like in crowdsourcing [53]. However, on a federated learning platform where task requesters and data providers can join and leave freely, task assignment can still be meaningful. This makes the platform similar to a data market [22]. But it still has unique challenges of FL, such as data privacy concerns, and the resource budgets of the data providers.

Acceleration of Encryption Schemes. The encryption schemes for privacy protection in FL such as homomorphic encryption and secret sharing often brings extremely large computation cost. For example, training a simple MLP on the MNIST dataset with homomorphic encryption can take 10 times slower than the original algorithm without any privacy protection [9]. The time consumption will be intolerable with larger dataset and deeper models like CNN. Therefore, how to accelerate the encryption schemes in FL has become a crucial problem.

Personalization. Most of existing works focus on training a global model rather than a personalized one for each client. With the help of multi-task learning, the participants can have personalized results by learning separate but related models [44]. However, it can only work with a small number of participants. In a large-scale cross-device scenario, novel domain adaptation techniques need to be designed to realize the full personalization.

Fairness. Some existing works already consider the fairness problem in FL [54, 55]. However, the problem is still challenging especially in scenarios with millions of mobile devices. Moreover, differential privacy is commonly used as a privacy preserving technique in FL but it may result in unfairness in model training [56]. How to compromise the strict constraints while still preserving the fairness in FL remains an open question.

General FL Platform. Like a crowdsourcing platform, a general FL platform can bring more opportunities to both data providers and task requesters. However, designing such a platform is challenging. The aforementioned key components should all be considered. The platform should support different types of learning algorithms and privacy preserving schemes in different scenarios. Also, it should provide incentive and pricing mechanisms for the participators. The communication cost and the model quality should also be optimized. Existing FL systems and benchmarks [57] only implement part of these features and cannot work as a general FL platform.

7 Conclusion

Federated Learning has gained much attention in recent years as a promising solution to the data isolation problem in artificial intelligence. Both as human-empowered AI techniques, federated learning and crowdsourcing have many similarities. In this article, we discuss four core issues in federated learning from the perspective of crowdsourcing, namely privacy and security, mechanism design, communication optimization and quality control. We find that the design of federated learning systems can be inspired by existing techniques in crowdsourcing platforms. We also envision some future directions of federated learning, which would be helpful to build a fully fledged federated learning platform.

References

[1] A. I. Chittilappilly, L. Chen, and S. Amer-Yahia, "A survey of general-purpose crowdsourcing techniques," *IEEE TKDE*, vol. 28, no. 9, pp. 2246–2266, 2016.

[2] Y. Tong, Z. Zhou, Y. Zeng *et al.*, "Spatial crowdsourcing: a survey," *VLDB*, vol. 29, no. 1, pp. 217–250, 2020.

[3] J. Deng, W. Dong, R. Socher *et al.*, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009, pp. 248–255.

[4] J. Konecný, H. B. McMahan, F. X. Yu *et al.*, "Federated learning: Strategies for improving communication efficiency," *CoRR*, vol. abs/1610.05492, 2016.

[5] B. McMahan, E. Moore, D. Ramage *et al.*, "Communication-efficient learning of deep networks from decentralized data," in *AISTATS*, vol. 54, 2017, pp. 1273–1282.

[6] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM TIST*, vol. 10, no. 2, pp. 12:1–12:19, 2019.

[7] Y. Tong, J. She, B. Ding *et al.*, "Online mobile micro-task allocation in spatial crowdsourcing," in *ICDE*, 2016, pp. 49–60.

[8] Y. Tong, Y. Zeng, B. Ding, L. Wang, and L. Chen, "Two-sided online micro-task assignment in spatial crowdsourcing," *IEEE TKDE*, 2019. [Online]. Available: doi.org/10.1109/TKDE.2019.2948863

[9] L. T. Phong, Y. Aono, T. Hayashi *et al.*, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE TIFS*, vol. 13, no. 5, pp. 1333–1345, 2018.

[10] C. Dwork, "Differential privacy," in *ICALP*, 2006, pp. 1–12.

[11] M. Abadi, A. Chu, I. J. Goodfellow *et al.*, "Deep learning with differential privacy," in *CCS*, 2016, pp. 308–318.

[12] Ú. Erlingsson, V. Feldman, I. Mironov *et al.*, "Amplification by shuffling: From local to central differential privacy via anonymity," in *SODA*, 2019, pp. 2468–2479.

[13] D. Jiang, Y. Song, Y. Tong *et al.*, "Federated topic modeling," in *CIKM*, 2019, pp. 1071–1080.

[14] Y. Wang, Y. Tong, and D. Shi, "Federated latent dirichlet allocation: A local differential privacy based framework," in *AAAI*, 2020, pp. 6283–6290.

[15] K. Bonawitz, V. Ivanov, B. Kreuter *et al.*, "Practical secure aggregation for privacy-preserving machine learning," in *CCS*, 2017, pp. 1175–1191.

[16] Z. Erkin, T. Veugen, T. Toft *et al.*, "Generating private recommendations efficiently using homomorphic encryption and data packing," *IEEE TIFS*, vol. 7, no. 3, pp. 1053–1066, 2012.

[17] P. Blanchard, E. M. E. Mhamdi, R. Guerraoui *et al.*, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *NIPS*, 2017, pp. 119–129.

[18] D. Alistarh, Z. Allen-Zhu, and J. Li, "Byzantine stochastic gradient descent," in *NIPS*, 2018, pp. 4618–4628.

[19] J. Steinhardt, P. W. Koh, and P. Liang, "Certified defenses for data poisoning attacks," in *NIPS*, 2017, pp. 3517–3529.

[20] M. Juuti, S. Szyller, S. Marchal *et al.*, "PRADA: protecting against DNN model stealing attacks," in *IEEE EuroSP*, 2019, pp. 512–527.

[21] Y. Tong, L. Wang, Z. Zhou *et al.*, "Dynamic pricing in spatial crowdsourcing: A matching-based approach," in *SIGMOD*, 2018, pp. 773–788.

[22] R. C. Fernandez, P. Subramaniam, and M. Franklin, "Data market platforms: Trading data assets to solve data problems," *PVLDB*, vol. 13, no. 11, pp. 1933–1947, 2020.

[23] S. R. Pandey, N. H. Tran, M. Bennis *et al.*, "A crowdsourcing framework for on-device federated learning," *IEEE TWC*, vol. 19, no. 5, pp. 3241–3256, 2020.

[24] J. Kang, Z. Xiong *et al.*, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE IoTJ*, vol. 6, no. 6, pp. 10 700–10 714, 2019.

[25] T. Song, Y. Tong, and S. Wei, "Profit allocation for federated learning," in *BigData*, 2019, pp. 2577–2586.

[26] J. Konecný, B. McMahan, and D. Ramage, "Federated optimization: Distributed optimization beyond the datacenter," *CoRR*, vol. abs/1511.03575, 2015.

[27] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *IEEE ICC*, 2019, pp. 1–7.

[28] H. Wang, Z. Kaplan, D. Niu *et al.*, "Optimizing federated learning on non-iid data with reinforcement learning," in *IEEE INFOCOM*, 2020, pp. 1698–1707.

[29] S. J. Reddi, Z. Charles, M. Zaheer *et al.*, "Adaptive federated optimization," *CoRR*, vol. abs/2003.00295, 2020.

[30] L. He, A. Bian, and M. Jaggi, "COLA: decentralized linear learning," in *NIPS*, 2018, pp. 4541–4551.

[31] M. Kamp, L. Adilova, J. Sicking *et al.*, "Efficient decentralized deep learning by dynamic model averaging," in *PKDD*, vol. 11051, 2018, pp. 393–409.

[32] A. Koloskova, S. U. Stich, and M. Jaggi, "Decentralized stochastic optimization and gossip algorithms with compressed

communication," in *ICML*, vol. 97, 2019, pp. 3478–3487.

[33] A. G. Roy, S. Siddiqui, S. Pölsterl *et al.*, "Braintorrent: A peer-to-peer environment for decentralized federated learning," *CoRR*, vol. abs/1905.06731, 2019.

[34] A. T. Suresh, F. X. Yu, S. Kumar *et al.*, "Distributed mean estimation with limited communication," in *ICML*, vol. 70, 2017, pp. 3329–3337.

[35] Y. Lin, S. Han, H. Mao *et al.*, "Deep gradient compression: Reducing the communication bandwidth for distributed training," in *ICLR*, 2018.

[36] H. Tang, S. Gan, C. Zhang *et al.*, "Communication compression for decentralized training," in *NIPS*, 2018, pp. 7663–7673.

[37] A. Reisizadeh, A. Mokhtari, H. Hassani *et al.*, "Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization," in *AISTATS*, vol. 108, 2020, pp. 2021–2031.

[38] S. Muthukrishnan, "Data streams: algorithms and applications," in *SODA*, 2003, pp. 413–413.

[39] W. Zhu, P. Kairouz, B. McMahan *et al.*, "Federated heavy hitters discovery with differential privacy," in *AISTAS*, S. Chiappa and R. Calandra, Eds., vol. 108, 2020, pp. 3837–3847.

[40] J. Jiang, F. Fu, T. Yang *et al.*, "Sketchml: Accelerating distributed machine learning with data sketches," in *SIGMOD*, 2018, pp. 1269–1284.

[41] N. Ivkin, D. Rothchild, E. Ullah *et al.*, "Communication-efficient distributed SGD with sketching," in *NIPS*, 2019, pp. 13 144–13 154.

[42] T. Li, Z. Liu, V. Sekar *et al.*, "Privacy for free: Communication-efficient learning with differential privacy using sketches," *CoRR*, vol. abs/1911.00972, 2019.

[43] P. Kairouz, H. B. McMahan, B. Avent *et al.*, "Advances and open problems in federated learning," *CoRR*, vol. abs/1912.04977, 2019.

[44] V. Smith, C. Chiang, M. Sanjabi *et al.*, "Federated multi-task learning," in *NIPS*, 2017, pp. 4424–4434.

[45] M. Yurochkin, M. Agarwal, S. Ghosh *et al.*, "Bayesian nonparametric federated learning of neural networks," in *ICML*, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97, 2019, pp. 7252–7261.

[46] H. Wang, M. Yurochkin, Y. Sun *et al.*, "Federated learning with matched averaging," in *ICLR*, 2020.

[47] X. Li, K. Huang, W. Yang *et al.*, "On the convergence of fedavg on non-iid data," in *ICLR*, 2020.

[48] Y. Zhao, M. Li, L. Lai *et al.*, "Federated learning with non-iid data," *CoRR*, vol. abs/1806.00582, 2018.

[49] E. Jeong, S. Oh, H. Kim *et al.*, "Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data," *CoRR*, vol. abs/1811.11479, 2018.

[50] N. H. Tran, W. Bao, A. Y. Zomaya *et al.*, "Federated learning over wireless networks: Optimization model design and analysis," in *IEEE INFOCOM*, 2019, pp. 1387–1395.

[51] S. Wang, T. Tuor, T. Salonidis *et al.*, "Adaptive federated learning in resource constrained edge computing systems," *IEEE JSAC*, vol. 37, no. 6, pp. 1205–1221, 2019.

[52] W. Y. B. Lim, N. C. Luong, D. T. Hoang *et al.*, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE ComSur*, vol. 22, no. 3, pp. 2031–2063, 2020.

[53] Y. Tong, L. Chen, Z. Zhou *et al.*, "SLADE: A smart large-scale task decomposer in crowdsourcing," *IEEE TKDE*, vol. 30, no. 8, pp. 1588–1601, 2018.

[54] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic federated learning," in *ICML*, vol. 97, 2019, pp. 4615–4625.

[55] T. Li, M. Sanjabi, A. Beirami *et al.*, "Fair resource allocation in federated learning," in *ICLR*, 2020.

[56] E. Bagdasaryan, O. Poursaeed, and V. Shmatikov, "Differential privacy has disparate impact on model accuracy," in *NIPS*, 2019, pp. 15 453–15 462.

[57] K. Bonawitz, H. Eichner, W. Grieskamp *et al.*, "Towards federated learning at scale: System design," in *MLSys*, 2019.

Human-in-the-loop Techniques in Machine Learning

Chengliang Chai, Guoliang Li
Department of Computer Science, Tsighua University
{chaicl15@mails.tsinghua.edu.cn, liguoliang@tsinghua.edu.cn}

Abstract

Human-in-the-loop techniques are playing more and more significant roles in the machine learning pipeline, which consists of data preprocessing, data labeling, model training and inference. Humans can not only provide training data for machine learning applications, but also directly accomplish some tasks that are hard for the computer in the pipeline, with the help of machine-based approaches. In this paper, we first summarize the human-in-the-loop techniques in machine learning, including: (1) Data Extraction: Non-structured data always needs to be transformed to structured data for feature engineering, where humans can provide training data or generate rules for extraction. (2) Data Integration: In order to enrich data or features, data integration is proposed to join other tables. Humans can help to address some machine-hard join operations. (3) Data Cleaning: In real world, data is always dirty. We can leverage humans' intelligence to clean the data and further induce rules to clean more. (4) Data Annotation and Iterative labeling. Machine learning always requires a large volume of high-quality training data, and humans can provide high quality data for training. When the budget is limited, iterative labeling is proposed to label the informative examples. (5) Model training and inference. For different applications(e.g. classification, clustering), given human labels, we have different ML techniques to train and infer the model. Then we summarize several commonly used techniques in human-in-the-loop machine learning applied in the above modules, including quality improvement, cost reduction, latency reduction, active learning and weak supervision. Finally, we provide some open challenges and opportunities.

1 Introduction

Machine learning (ML) has seen great success on a wide variety of applications, such as image and speech recognition, natural language processing and heath care. It has made breakthroughs due to large-scale data, high computing power, and sophisticated algorithms, but the power of humans cannot be neglected, where large-scale training data needs humans to create and some ML algorithms require humans to improve the performance iteratively. For example, ImageNet [12] is a representative benchmark that promotes the development of computer vision area. It is constructed through crowdsourcing, which is an effective way to address a wide variety of tasks by utilizing hundreds of thousands of ordinary workers (e.g., humans).

Humans play important roles in the entire ML pipeline from data preparation to result inference, as shown in Figure 1. Before building a model, data scientists spend more than 80% of their time in preprocessing the data [1],

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

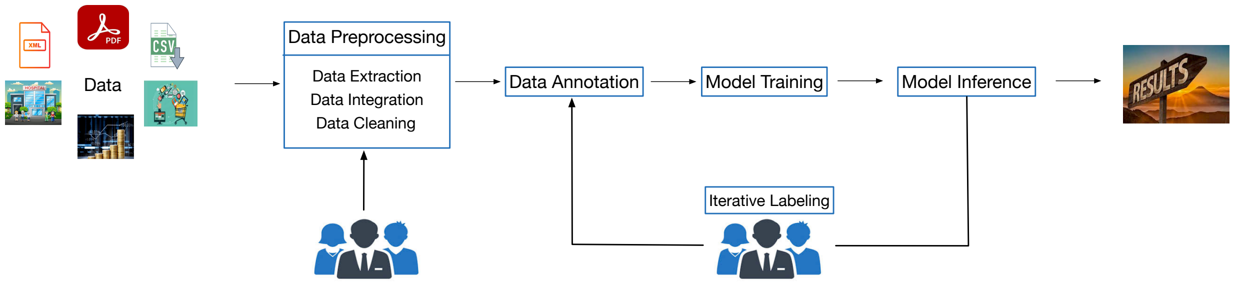


Figure 1: A Human-in-the-loop Machine Learning Pipeline

including data extraction, data integration and data cleaning. Then data is labeled and divided into training and test sets. Finally we train and test the model. Humans can contribute to all steps mentioned above.

(1) Data Extraction. In most cases, original data we can utilize to build a model may be unstructured or semi-structured, which needs to be extracted as structured data to construct features. Data extraction is to use rules (functions) or machine learning techniques [36, 43, 15] to extract data from non-structured data, where humans can provide rules or training data.

(2) Data Integration. Given the structured data from multiple sources, we always have to integrate them [9, 61, 59] to enrich the records and features. Data integration is used to identify duplicated records or cells in different columns that refer to the same entity, such as “Apple iPhone 8” and“iPhone 8th”, and then integrate them. Humans can improve the performance of data integration by providing answers of entity pairs that are hard for the computer. Also, ML techniques can also be used to address the problem, where humans can provide training data. In addition, before integrating the data, we should align the columns of different relational tables, i.e., schema matching. We can leverage human intelligence as well as knowledge bases to identify matching schemes, which are hard for the computer.

(3) Data Cleaning. In the real world, data is always dirty because of some missing values, duplicates, outliers and records that violate integrity constraints [59, 7, 10], so data cleaning can detect and repair these data, which are likely to improve the ML performance. For different data cleaning tasks, we can leverage humans’ cognitive ability to address them. For example, for duplicates, one can leverage machine to identify easy duplicated records and left the hard ones to humans [59].

(4) Data Annotation and Iterative Labeling. Each record has to be labeled to construct the training set. Humans can provide high quality labels directly. Then the model is trained and tested on the labeled data. However, since humans are not free, requiring large quantities of labels is expensive. Therefore, humans will be asked to label the most interesting examples iteratively until a good performance is achieved.

(5) Model training and inference. For different machine learning tasks, like classification and clustering, there are different techniques that leverage humans’ labels to train and infer the results. For classification, one can utilize techniques like deep learning, expectation maximization or graph model to deduce the results based on noisy human labeled data. For clustering, a straightforward method is to leverage a human-machine hybrid method to cluster these examples(e.g. video or images) that are hard to cluster purely by computers. In addition, generative models can also be applied to cluster examples according to multiple criteria.

Although humans contribute to different modules in the ML pipeline, there are several common important problems in human-in-the-loop machine learning. The first is quality improvement, Humans are likely to make mistakes no matter what kinds of tasks they do because they may have different levels of expertise, and an untrained human is not qualified to accomplish certain tasks. To achieve high quality labels, we need to tolerate human errors and infer high quality results from noisy answers. Secondly, Since humans are not free, if there are large numbers of tasks, it is expensive to leverage humans to address all of them. Therefore, several techniques are proposed such as pruning, answer deduction and sampling. Thirdly, generally speaking, humans are much slower than the computer. To accomplish the tasks efficiently, we should reduce the latency, which is the time from the user submits the first task to the final answer is returned. Fourthly, given a task, a user does not always have enough budget to label a large number of training data. Therefore, active learning is proposed to involve humans to label the most interesting examples iteratively so that the examples in each iteration affect the model as much as possible. Lastly, in active learning, we assume the labels provided by humans are perfect, but it does not hold in reality. Therefore, weak supervision is proposed to obtain a relative high quality result through a large number of weak labels, which are provided by humans with different qualities or functions (rules).

In a word, we will introduce what humans can contribute in the machine learning pipeline in Section 2. Next, several significant human-in-the-loop techniques are introduced in Section 3. Then we discuss some open challenges and opportunities in Section 4 and conclude in Section 5.

2 Human-in-the-loop Machine Learning Pipeline

As shown in Figure 1, humans play significant roles in machine learning pipeline. First, given some unstructured data, we have to transform it structured data, in order to construct features for ML. Then for structured data from multiple sources, we should integrate them for enriching data and features to achieve well-performed ML model. What's more, data is always dirty in the real world. To further improve the performance, we should clean the data, such as repairing records that violate integrity constraint and removing outliers and duplicates. Finally, we should annotate the data for building the model. For all above steps in the pipeline, humans can contribute their intelligence to provide high quality training data and improve the ML model. Next, we will introduce what humans can contribute in these steps.

2.1 Data Extraction

Extracting structured data from unstructured data is an important problem both in industry and academia, which has been studied broadly from rule-based [36] systems to ML-based approaches [43, 15]. However, these methods either need domain experts to design rules or humans to provide large quantities of labels. Recently, DeepDive [69] is a representative system in this area, which provides declarative language for non-expert users to extract data. The execution of DeepDive can be divided into three parts: candidate generation, supervision, statistical inference and learning. Humans mainly contribute in the first part, i.e., candidate generation. In this part, humans write some extraction rules described by declarative languages to retrieve data with attributes or relations, such as entity B is the wife of A if there exists mention "and his wife" between A and B in a corpus. The goal of this part is to generate candidates with high recall and low precision. Secondly, the supervision part applies distant supervision rules from knowledge bases or incomplete databases to provide labels for some of the candidates. The rules do not need to label all candidates from the first part, which are intended to be a low recall and high precision. For the last part, DeepDive constructs a graphical model that represents all of the labeled candidate extractions, trains the model, and then infers a correct probability for each candidate. At the end of this stage, DeepDive applies a threshold to each inferred probability and then derives the extractions to the output database. In conclusion, Deepdive leverages humans to provide extraction candidates with high recall, uses weak supervision(distance supervision) to label them and finally trains a statistical ML model to fine-tune the labels.

2.2 Data Integration

Given relational tables from multiple sources, in many cases we want to integrate them for extending existing datasets, including features and records. To this end, schema matching [70, 17] and entity resolution [59, 61] have to be applied, where the first part is going to align the columns and the second will match records from different tables. Recently, many existing works focused on leveraging human intelligence to achieve these.

For schema matching, existing works [70] utilize human-machine hybrid approaches to improve the performance. They utilize machine-based schema matching tools to generate a set of possible matchings, each of which has a probability to be matched. They define a correspondence correctness question (CCQ) for humans to answer, which denotes a pair of attributes from two columns, so each matching consists of a set of correspondences. Then the problem is to wisely choose the correspondences to ask the human to obtain the highest certainty of correct schema matching at the lowest cost. The uncertainty is measured by entropy on top of the probabilities that the tools generate. In the correspondence selection, they consider the column correlations, selection efficiency and human quality to match schemes effectively and efficiently. Fan et.al [17] introduce knowledge base together with humans to do schema matching. First, they propose a concept-based approach that maps each column of a table to the best concept in knowledge bases. This approach overcomes the problem that sometimes values of two columns may be disjoint, even though the columns are related, due to incompleteness in the column values. Second, they develop a hybrid machine-crowdsourcing framework that leverages human intelligence to discern the concepts for “difficult” columns. The overall framework assigns the most “beneficial” column-to-concept matching tasks to the human under a given budget and utilizes the answers to infer the best matching.

After the schemes are aligned, we can integrate different relational tables by the join operation. Traditionally, join is always executed by exact matching between values of attributes from two tables. However, in the real world, data is always dirty. For example, “Apple iPhone 8” and ”iPhone 8th” refer to the same entities and should be joined, which cannot be done by a traditional database. Therefore, the human-based join is proposed to address this problem. Wang et.al. [59] propose crowd-based join framework, which generates many candidate pairs, uses similarity based pruning techniques to eliminate dissimilar pairs and ask the crowd to answer the rest pairs. To further reduce the cost, Wang et.al. [61] leverage the transitivity technique to deduce unknown answers based on current answers from humans. Chai et.al. [9, 8] build a partial-order graph based on value similarities of different attributes and utilize the graph to prune pairs that are not necessary to ask. To improve the quality, Wang et.al. [62] first cluster the entities to be joined and then leverage humans to refine the clusters. Yalavarthi et.al. [67] select questions judiciously considering the crowd errors.

2.3 Data Cleaning

Data is dirty in the real world, which is likely to hurt the ML performance. For example, some values may be out of range (e.g., age is beyond 120 or below 0) or utilize wrong units (e.g., some distances are in meters while other are in kilometers); Some records refer to the same entity; Integrity constraints (e.d. functional dependencies) are violated among records. Recently, many researchers focused on leveraging human to clean the data. For instance, crowd-based entity resolution [59, 9, 8] is applied to remove duplicates. Chai et.al. [7] use human expertises to identify outliers among the data. Specifically, they first utilize machine-based outlier detection algorithms to detect some outlier candidates as well as inlier candidates, and then human is asked to verify these candidates by comparing outlier candidates with inliers. Chu et.al. [10] clean the data that violates integrity constraints with the help of knowledge base and humans. They first identify the relationships between columns using knowledge base and then use humans to verify them. Then the discovered relationships can be utilized to detect errors among data, and then these error can be repaired by the knowledge base and humans.

Recently, a line of interesting data cleaning works focus on cleaning with the explicit goal of improving the ML results. Wang et al. [29] propose a cleaning framework ActiveClean for machine learning tasks. Given a dataset and machine learning model with a convex loss, it selects records that can most improve the performance of the model to clean iteratively. ActiveClean consists of 4 modules, sampler, cleaner, updater and estimator. Sampler is used to select a batch of records to be cleaned. The selection criterion is measured by how much improvement can be made after cleaning a record, i.e., the variation of the gradient, which is estimated by the Estimator. Then the selected records will be checked and repaired by the Cleaner, which can be humans. Next, the Updater updates the gradient based on these verified dirty data. The above four steps are repeated until the budget is used up. BoostClean [30] cleans the data where an attribute value is out of range. It takes as input a dataset and a set of functions for detecting errors and repair functions. These functions can be provided by humans. Each pair of detection and repair functions can produce a new model. BoostClean uses statistical boosting to find the best ensemble of pairs that maximize the final performance. Recently, TARS [13] was proposed to clean human labels using oracles, which provides two pieces of advice. First, given test data with noisy labels, TARS estimates the performance of the model on true labels, which is shown to be unbiased and confidence intervals are computed to bound the error. Second, given training data with noisy labels, TARS determines which examples to be sent to an oracle so as to maximize the expected model improvement of cleaning each noisy label.

2.4 Iterative Labeling

After the above steps of data preprocessing, we can label the data in relational tables for ML tasks. The most straightforward method is to directly leverage humans to annotate a bunch of data for training. Thus we can adopt the cost control and quality control approaches proposed in Section 3 to derive high quality labels with low cost (see [32] for a survey). However, in many cases, a user does not have enough budget to obtain so many annotations. Therefore, many researchers focused on how to label data iteratively and make the model performance better and better using techniques like active learning or weak supervision.

Mozafari et.al. [42] use active learning to scale up the human labeling, which can be utilized in two scenarios, the upfront and iterative scenario. In the upfront scenario, the user cares more about the latency than the cost. Therefore, given a budget and an initial model, the algorithm uses a ranker to rank and selects some of the most informative examples to label while the rest are predicted by the model. In the iterative scenario, since the user cares more about the cost, the ranker selects a batch of examples to label, retrains the model and selects again until the budget is used up. There are two strategies (Uncertainty and MinExpError) that the user can choose for ranking. Leveraging the traditional active learning technique, Uncertainty selects examples that the current model is the most uncertain about. MinExpError uses a more sophisticated algorithm that considers both the uncertainty and expected model change. Besides, the work also utilizes the bootstrap theory, which makes the algorithms available to any classifier and also enables parallel processing. Also, active learning techniques in section 3.4 can also be integrated in the framework.

DDLite [14] leverage human to conduct data programming rather than hand-labeling data, in order to generate large quantities of labels. Given a set of input documents, DDLite aims to produce a set of extracted entities or relation mentions, which consists of four steps. First, given input documents, preprocessing like domain-specific tokenizers or parsers of the raw text has to be performed. Second, DDLite provides a library of general candidate extraction operators, which can be designed by humans. Third, humans develop a set of labeling functions through iterating between labeling some small subsets and analyzing the performance of labeling functions. Lastly, features are automatically generated for the candidates, and then the model is trained using the labeling functions. The humans then analyze the performance on a test set.

2.5 Model Training and Inference

For different machine learning tasks, there are different techniques that leverage humans' knowledge to train and infer the results, considering humans' diverse qualities. In this part, we mainly discuss two common ML tasks, classification and clustering, and show how to leverage human intelligence as well as ML techniques to achieve high quality results.

For classification, it is expensive to obtain reliable labels to train a model, so multiple humans are required to collect subjective labels. Raykar et.al. [48] first proposed a straightforward method that simply utilizes majority voting(MV) to infer labels. However, MV does not consider features of examples. Therefore, given the human labels and features of examples, they improve the model by considering the true labels as latent variables and utilize the Expectation-Maximization (EM) algorithm to train the model. The parameters include the worker qualities and feature weights. Rodrigues et.al. [50] also use EM algorithm to jointly learn the parameters of humans and examples. The difference is that they use deep learning to train the model, where a crowd layer is proposed to allow the neural network to learn directly from noisy humans labels in an end-to-end manner. In some cases, acquiring large quantities of labels is expensive, so Atarashi et.al. [3] proposed to learn from a small number of human labels and unlabeled data using deep generative model in a semi-supervised way. More specifically, they leverage the unlabeled data effectively by introducing latent features and a data distribution. Because the data distribution can be complex, they use a deep neural network for the data distribution. Classification based on taxonomy is a particular but important task that the labels can consist of a taxonomy. For example, BMW X3 and BMW X5 belong to BMW, which belongs to Car. For this scenario, Parameswaran et.al. [45] utilize a human-machine hybrid method to classify the examples on the taxonomy. For example, given a picture of an Audi car, we can ask the humans to label whether it is a BMW car. If not, the children of BMW (BMW X3 and BMW X5) can be pruned. They study how to use the minimum number of questions to get all the labels.

For clustering, we can also leverage human intelligence to cluster examples that are hard to identify by computers. Following the k-means algorithm, Heikinheimo et.al. [23] propose a human-in-the-loop framework that asks the humans to answer a simple task each time and aggregate all the answers to deduce the final clustering result. Specifically, the simple task is, given a triple with three objects, asking the human to select the one different from the other two objects. First, the algorithm picks a large enough number of triplets from the entire dataset and asks humans to label them. Second, for each example, they compute a penalty score defined as the number of times the example was chosen to be different. Third, the example having the lowest penalty score is returned. Thus, the centroid example of each cluster is computed and we can obtain the clustering results iteratively. However, this method is expensive because of the large number of triple tasks and cannot generalize when there are new examples. To address this, Gomes et.al. [20] propose to use a generative model to infer the clusters. Moreover, it can capture multiple clustering criteria from diverse viewpoints of humans. For example, given a set of pictures of products, one may want to cluster by brands while another human is likely to cluster by types. Specifically, they divide the entire set into small groups and ask humans to cluster examples in each group. Then considering the humans' quality and labels, [20] uses a Bayesian generative model to infer the clustering results.

3 Human-in-the-loop Techniques for Machine Learning

As shown in Section 1, humans are involved frequently and necessarily in a machine learning pipeline. They can not only contribute to the data preprocessing steps, but also provide a large amount of labeled training data to build a well-performed machine learning model, especially for the deep learning [12]. No matter what roles humans play in a ML pipeline, there exist some common sophisticated techniques to apply. In this section, we summarize some significant techniques in human involved machine learning. First, when humans are asked to conduct data annotation or data preprocessing, they are always required to provide high quality results, so we should study how to improve the quality of human answers in section 3.1. Second, since humans are not free, we study how to save monetary cost while not sacrificing much quality in section 3.2. Third, since humans cannot perform as quickly as machines, latency should be reduced to accelerate the entire ML process(section 3.3). Besides, active learning(section 3.4) focuses on selecting the most interesting examples to human for labeling to improve the model iteratively, which is an advanced technique in the field of machine learning. Lastly, in section 3.5, we discuss the situation where the user cannot derive a number of high quality labels, so she has to use weak supervision techniques to build a model based on weak labels, still with satisfying performance.

3.1 Quality Improvement

Human answers may not be reliable because (1) there exist malicious humans that randomly return answers, especially in the crowdsourcing scenario and (2) some tasks are difficult for humans to answer. Therefore, it is significant to discover different characteristics of humans and tasks, which can be leveraged to improve the quality. There are two commonly-used techniques for quality improvement, i.e., truth inference and task assignment, which will be introduced as follows.

Truth Inference. To control the quality, an intuitive idea is to assign each task to multiple humans, aggregate the answers and infer the truth. Note that humans may provide low quality or even malicious answers because they may also have different levels of expertise, and an untrained human may be incapable of answering certain tasks. Therefore, to achieve high quality, we need to tolerate human errors and infer high-quality results from noisy answers.

A unified quality control framework consists of the following three steps. First, we initialize each human's quality. Second, we infer the truth based on the collected answers and current quality. Third, we estimate the quality according to the inferred truth. Then we iterate the second and third steps until converge. Based on the unified framework, existing works [39, 64] can be categorized based on the following three factors: task modeling, human modeling and applied techniques(how to use task and human modeling to infer the truth).

(1) Task Modeling. This describes how existing solutions model a task, mainly including the difficulty of a task and the latent topics in a task [39, 64]. First, some recent works model the difficulty levels of a task instead of assuming that a human has the same quality for answering all the tasks. The more difficult a task, the harder a human can provide a perfect answer for it. For example, in [64], $Pr(v_i^w = v_i^* | d_i, q^w) = 1/(1 + e^{d_i q^w})$ denotes the probability that human w correctly answers task t_i, where $d_i \in (0, +\infty)$ represents the difficulty of task t_i, v_i^w is the worker's answer for a task v_i whose true answer is v_i*, q_w is the worker quality. The higher d_i, the easier task the t_i . Intuitively, for a fixed human quality q^w >0, an easier task (high value of d_i) leads to a higher probability that the human correctly answers the task. Second, some recent works model the difficulty as a vector with K values instead of a single value. The basic idea is to exploit diverse topics of a task, where K is the pre-defined number of topics. For example, existing works [16, 39] apply topic model techniques on text description of each task to derive the topic vector. Besides, entity linking techniques are utilized to infer the topic vector for each task [75].

(2) Human Modeling. This describes how existing works model a human's quality, which is always denoted as a single real number $q^w \in [0, 1]$, representing the probability that human w answers a task correctly. This straightforward model has been widely adopted by existing works [37, 11, 4]. More specifically, for single-choice tasks, existing works [57, 28, 48] extend the above model to the confusion matrix to model the human quality in a more fine-grained way. Suppose each task in has l fixed choices, then the confusion matrix q^w is an ll matrix, where the j-th ($1 \leq j \leq l$) row, i.e., $q_j^w = [q_{j,1}^w, q_{j,2}^w, ..., q_{j,l}^w]$, represents the probability distribution of human w's possible answers for a task if the truth of the task is the j-th choice. Each element $q_{j,k}^w$ denotes that given the truth of a task is the j-th choice, the probability that human w selects the k-th choice. For numeric tasks, human bias and variance are proposed to model the human quality [48, 63]. Bias measures the effect that a human may underestimate (or overestimate) the truth of a task and variance measures the variation of errors around the bias. What's more, existing works [25, 34] introduce confidence in quality control, i.e., if a human answers many tasks, then the estimated quality for her is of high confident; otherwise the estimated quality is not confident. Inspired by this , [34] assigns higher qualities to the humans who answer plenty of tasks.

(3) Applied Techniques. In this part, we discuss how existing works leverage task models and human models to solve the truth inference problem. In general, existing works adopt the aforementioned unified framework, which can be categorized as the following three classes: straightforward computation [19, 44], optimization methods [4, 34, 35, 77] and probabilistic graphical model methods [27, 37]. First, the straightforward computation are some baseline models that estimate the truth without modeling the human or tasks. For single-label tasks, they always use the majority voting to address. For numerical tasks, mean and median are two baseline methods that regard the mean and median of humans' answers as the truth. Second, optimization methods focus on designing optimization functions that capture the relations between humans' qualities and tasks' truth, and then provide an iterative method to compute these two sets of parameters. The differences among existing works [4, 34, 35, 77] are that they model humans' qualities based on the above human modeling part differently. Third, probabilistic

graph models a human's quality as a node and utilize graphical model inference to iteratively derive humans' models [27, 37], where a graphical model is a graph, containing nodes and edges between pairs of nodes. Each node represents a random variable, which can be unknown parameters or observed data, and each edge represents the possible relationship (e.g., conditional dependency) between the linked pair of nodes.

Task Assignment. Since humans diverse backgrounds and qualities on tasks, a sophisticated task assignment algorithm will judiciously select tasks to right humans. Existing works mainly focus on two scenarios: (1) human-based, i.e., given a task, which subset of humans should be selected to answer the task; (2) task-based, i.e., when a human comes, which subset of tasks should be assigned to the human.

(1) Human-based. In this scenario, given a task and a set of candidate humans, the focus is on studying which subset of humans should be selected to answer the task in order to maximize the task's quality without exceeding the overall budget. The problem is often called the "Jury Selection Problem" [6, 74]. Intuitively, humans with high quality should be selected. To this end, Cao et al. [9] provide a framework that first studies how to compute the quality of a given subset of humans before they give answers, called Jury Quality (JQ). Since the answers are unknown in advance, all possible cases of humans' answers should be considered to compute the quality. To address this, Cao et al. [6] propose a Majority Voting strategy to compute the JQ. Zheng et al. [74] prove that Bayesian Voting is the optimal strategy under the definition of JQ. That is, given any fixed subset of humans S, the JQ of S w.r.t. the Bayesian Voting strategy is not lower than the JQ of S w.r.t. any other strategy. Therefore, given a set of humans, its JQ w.r.t. Bayesian Voting strategy is the highest among all voting strategies.

(2) Task-based. In this scenario, when a human comes, the focus is on studying which subset of tasks should be assigned to the coming human. This problem is often called the "Online Task Assignment Problem". When a human comes, [38, 5] compute an uncertainty score for each task based on collected answers, select the k most uncertain tasks, and assign them to the human. There are multiple methods to define the uncertainty. Liu et al. [38] use a quality-sensitive answering model to define each task's uncertainty, and Boim et al. [5] leverage an entropy-like method to compute the uncertainty of each task. Besides, some other works [72, 73] model humans to have diverse skills among different domains, and choose the tasks from the domains that a coming human is good at to assign. What's more, many machine learning techniques [24, 76, 42] aim to assign a set of tasks to workers that are most beneficial to their trained models.

3.2 Cost Reduction

Humans are not free. Even if we turn to some cheap resources, like crowdsourcing for help to address the work, it can be still very expensive when there are a large number of tasks. Therefore, how to reduce the cost without sacrificing the quality is a big challenge. In this part, we introduce four kinds of techniques to reduce the human costs.

Pruning. Given a large number of tasks, pruning means that the user can conduct some preprocessing operations on then, so that some tasks are not necessary to be checked by humans. The basic idea is that some easy tasks can be addressed by the computer while the hard ones are left to humans. Pruning has been widely adopted in the area of human-powered join [11, 59, 61, 9, 8] an selection [68]. For example, the crowdsourcing join asks the human to identify records that refer to the same entity in the real world. To this end, the machine can compute a string similarity score for each pair of entities. Intuitively, those entities with very low(high) score are likely to be non-matching(matching) pairs, which can be easily solved purely by machine. For the rest hard ones, we can turn to the human for help. The advantage of this technique is that it is very straightforward, easy to implement and effective in many scenarios. However, the risk is that those pruned tasks cannot be checked by human, which may incur noise. Also, the threshold of deciding which part to prune is difficult to set.

Task Selection. Task selection has been introduced in section 3.1 for quality improvement. From another point of view, task selection can be seen as minimizing the human cost with a quality constrain. Different applications need different task selection strategies, such as join [11, 59, 61, 9, 8], top-k/sort [21, 33] categorize [45], etc. The basic idea is that given a task, a task selection strategy is first used to judiciously select a set of most beneficial tasks. Then after these tasks are sent to a platform with humans, a task assignment strategy is then used to collect high-quality answers from them. In a word, the task selection can achieve a good trade-off between cost and quality, especially the cost saving under a quality requirement. However, the downside is that it will incur much latency because the tasks are sent out iteratively.

Answer Deduction. Answer deduction can be adopted when the given tasks have some inherent relationships, which can be utilized to reduce the cost. Specifically, given a set of tasks, after deriving some results from humans, we can use this information to deduce some other tasks' results, saving the cost of asking the crowd to do these tasks. Many operators have such property, e.g., join [61, 9, 8], planning [26, 71], mining [2]. For example, suppose a join operator generates three tasks: (A, B), (B, C), and (A, C). If we have already known that A is equal to B, and B is equal to C, then we can deduce that A is equal to C based on transitivity, thereby avoiding the crowd cost for checking (A, C).

Sampling. A sampling-based technique only utilizes the humans to process a sample of data and then leverage their answers on the sample to deduce the result on the entire data. This technique has been shown to be very effective in human-powered aggregation [40], and data cleaning [119]. For example, Wang et al. [60] propose a sample-and-clean framework that allows the human to only clean a small sample of data and uses the cleaned sample to obtain high-quality results from the entire data.

3.3 Latency Reduction

Given all tasks submitted by a user, latency denotes the time until all tasks have been accomplished. Since humans need time to think and answer, they will be much slower than the machine, so it is necessary to reduce the latency. Existing approaches can be categorized into the round-based model and statistical model.

Round-based Model. In some cases, tasks are answered in multiple rounds. In each round, we can utilize task selection techniques to select a bunch of tasks. For these tasks, we can leverage multiple humans to answer them so that the latency can be reduced. Concretely, suppose there are enough humans, some existing works [52, 58] simplify the definition of latency by assuming that each round spends 1 unit time, and then the latency is modeled as the number of rounds. They use the round model to do latency control. To this end, answer deduction is applied to reduce the number of tasks. More specifically, tasks that do not have relationships will be asked in parallel in a single round, so that some answers of other tasks can be deduced without any more costs. Therefore, since the total number of tasks can be reduced, the latency will be reduced.

Statistical Model. Some existing works [68, 18] utilize statistics information from real crowdsourcing platforms to model workers' behaviors. Yan et al. [68] build statistical models to predict the time of answering a task, which considers (1) delay for the arrival of the first response; (2) the inter-arrival times between two responses. Faradani et al. [18] leverage statistical models to predict worker's arrival rate in a crowdsourcing platform and characterize how workers select tasks from the platform.

3.4 Active Learning

Active learning is a commonly used technique in machine learning, which involves humans to label the most interesting examples iteratively. It always assumes that humans can provide accurate answers. The key challenge is that given a limited budget, how to select the most appropriate examples in each iteration. Active learning has been extensively discussed in surveys [31, 49], so we only cover the most prominent techniques in this part.

Uncertainty sampling. Uncertainty sampling [31] is one of the simplest and commonly used methods in active learning, which selects the next unlabeled example which the current model regards as the most uncertain one. For example, when using a probabilistic model for binary classification, uncertainty sampling chooses the example whose probability is the close to 0.5. If there are more than three labels, a more general uncertainty sampling variant should be query the example whose prediction is the least confident. However, this approach throws away the information of other possible labels. Therefore, some researchers propose the marginal sampling, which chooses the example whose probability difference between the most and second likely labels is the smallest. This method can be further generalized by introducing the entropy for measuring the uncertainty.

Query-by-committee (QBC). The QBC [56] approach extends uncertainty sampling by maintaining a committee of models which are trained on the same labeled data. Each committee member can vote when testing each example, and the most informative example is considered to be the one where most models disagree with each other. The fundamental idea is to minimize the version space, which is the space of all possible classifiers that give the same classification results as the labeled data.

Expected model change. Another general active learning framework utilizes the decision-theoretic approach, choosing the example that would introduce the greatest change to the current model with the assumption that the label is known. A strategy of this framework is the "expected gradient length" (EGL) approach [55] for a discriminative probabilistic model, which can be applied to any learning problem where gradient-based training is used. In EGL, the change to the model can be measured as the length of training gradient. In other words, we should select the example that will lead to the largest gradient if it is labeled. However, since the true label is not known, we should compute the length of training gradient as an expectation over possible labels.

Expected error reduction. Another decision-theoretic method [51] aims to measure how much its generalization error is likely to be reduced rather than how much the model is likely to change. Given an example, the basic idea is to first estimate the expected future error of the model trained using the example together with current labeled data on the remaining unlabeled examples. Then the example induced the smallest error is selected. Similar to the EGL method, since we do not know the true label of each unlabeled example, the expectation of future error over all possible labels should be computed.

Density-weighted methods. The mentioned frameworks above are likely to choose the outlier examples, which might be uncertain and disagreeing but not representative. However, most time the outliers contribute less than the representative examples which follow the similar distribution of the entire dataset. Therefore, existing works [54, 65] focus on choosing examples not only uncertain or disagreeing, but also representative of the example distribution.

3.5 Weak Supervision

In the above section, active learning approaches always involve experts without generating noise into the machine learning iterations. However, some real applications always need a large number of training labels and asking experts to do so heavy work is expensive. Therefore, existing works [46, 15, 41] have focused on the weak supervision, which generates large amount of labels semi-automatically. These labels are not perfect but good enough to result in a reasonably-high accuracy. Next we summarize two techniques with respect to the weak supervision.

Data programming. Data programming [47] has been proposed to generate a large number of weak labels using multiple labeling functions rather than labeling for each example. Each function can be written by the human and the Snorkel system [46] provides a friendly interface to support it. Obviously, a single function is not effective enough to derive a well-performed model, so multiple functions should be combined to generate labels. The most straightforward method of combination is majority voting, but it does not consider the correlations and qualities of different functions. To address this, Snorkel [46] proposed a probabilistic graphical model to generate the weak labels which is followed by a discriminative model trained on the weak labels.

Fact extraction. Fact extraction is another way to generate weak labels using knowledge base, which contains facts extracted from different sources including the Web. A fact usually describes entities with attributes and relations, such as <`China`, `capital`, `Beijing`>, which indicates the capital of China is Beijing. The facts can be regarded as labeled examples, which can be used as seed labels for distant supervision [41]. Besides fact extraction can also be considered as extracting facts from multiple resources to construct a knowledge base. The Never-Ending Language Learner (NELL) system [15] continuously extracts structured information from the unstructured Web and constructs a knowledge base. Initially, NELL starts with seeds that consist of an ontology of entities and relationships among them. Then NELL explores large quantities of Web pages and identifies new entities pairs, which has the same relationships with seeds based on the matching patterns. The resulting entity pairs can then be used as the new training data for constructing even more patterns. The extraction techniques can be regarded as distant supervision generating weak labels.

4 Open Challenges and Opportunities

Data discovery for ML. Suppose an AI developer aims to build an ML model. Given a dataset corpus, the user requires to find relevant datasets to build the model. Data discovery aims to automatically find relevant datasets from data warehouse considering the applications and user needs. Many companies propose data discovery systems, like Infogather [66] in Microsoft and Goods [22] in Google. However, such systems focus on keyword-based dataset search or just linking datasets. Therefore, it may be worth studying to discover datasets that can directly maximize the performance of the downstream ML model. The key challenges lie in how to find valuable features and data among the corpus.

Modules selection in ML pipeline. Figure 1 shows the standard ML pipeline from data preparation to the model training and testing, which consists of several modules like schema matching, data cleaning and integration, etc, and data cleaning can also be extended to many scenarios, like missing values, outliers and so on. Given an ML task, asking the humans to process all modules is expensive, and it may not be necessary. Thus, we can study which modules are significant to the ML model and drop the other ones. For example, given a classification task, some data cleaning tasks like removing duplicates are not necessary. Therefore, how to select modules to optimize the ML pipeline is worth to study.

Trade-off between human quality and model performance. Some existing works [46, 15, 41] focus on acquiring weak labels to derive a model with good performance. One idea is to study the trade-off between human quality and model performance. That is, given a performance requirement, such as 80% F-measure, we can decide how to select humans to label the training data with the goal of optimizing the cost. Also, given human qualities, we can study how to produce results with the highest quality.

Benchmark. A large variety of TPC benchmarks (e.g., TPC-H for analytic workloads, TPC-DI for data integration) standardize performance comparisons for database systems and promote the development of the database community. Even though there are some open datasets for crowdsourcing or machine learning tasks, there is still lack of standardized benchmarks that covered the entire human involved machine learning pipeline. To better explore the research topic, it is significant to study how to develop evaluation methodologies and benchmarks for the human-in-the-loop machine learning system.

5 Conclusion

In this paper, we review existing studies in human-in-the-loop techniques for machine learning. We first discuss how to apply human-in-the-loop techniques to the ML pipeline including data extraction, data integration, data cleaning, labeling and ML training/inference. Then we introduce five commonly used techniques in this field, including quality improvement, cost reduction, latency reduction, active learning and weak supervision. Finally, we provide open challenges and opportunities.

References

[1] https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/.

[2] Y. Amsterdamer, S. B. Davidson, T. Milo, S. Novgorodov, and A. Somech. OASSIS: query driven crowd mining. In *SIGMOD 2014*, pages 589–600.

[3] K. Atarashi, S. Oyama, and M. Kurihara. Semi-supervised learning from crowds using deep generative models. In S. A. McIlraith and K. Q. Weinberger, editors, *AAAI 2018*, pages 1555–1562.

[4] B. I. Aydin, Y. S. Yilmaz, Y. Li, Q. Li, J. Gao, and M. Demirbas. Crowdsourcing for multiple-choice question answering. In C. E. Brodley and P. Stone, editors, *AAAI, 2014*, pages 2946–2953.

[5] R. Boim, O. Greenshpan, T. Milo, S. Novgorodov, N. Polyzotis, and W. C. Tan. Asking the right questions in crowd data sourcing. In *(ICDE 2012*, pages 1261–1264.

[6] C. C. Cao, J. She, Y. Tong, and L. Chen. Whom to ask? jury selection for decision making tasks on micro-blog services. *Proc. VLDB Endow.*, 5(11):1495–1506, 2012.

[7] C. Chai, L. Cao, G. Li, J. Li, Y. Luo, and S. Madden. Human-in-the-loop outlier detection. In *SIGMOD 2020*, pages 19–33.

[8] C. Chai, G. Li, J. Li, D. Deng, and J. Feng. Cost-effective crowdsourced entity resolution: A partial-order approach. In *SIGMOD 2016*, pages 969–984.

[9] C. Chai, G. Li, J. Li, D. Deng, and J. Feng. A partial-order-based framework for cost-effective crowdsourced entity resolution. *VLDB J.*, 27(6):745–770, 2018.

[10] X. Chu, M. Ouzzani, J. Morcos, I. F. Ilyas, P. Papotti, N. Tang, and Y. Ye. KATARA: reliable data cleaning with knowledge bases and crowdsourcing. *Proc. VLDB Endow.*, 8(12):1952–1955, 2015.

[11] G. Demartini, D. E. Difallah, and P. Cudré-Mauroux. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In A. Mille, F. L. Gandon, J. Misselis, M. Rabinovich, and S. Staab, editors, *WWW 2012*, pages 469–478.

[12] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li. Imagenet: A large-scale hierarchical image database. In *CVPR 2009*, pages 248–255.

[13] M. Dolatshah, M. Teoh, J. Wang, and J. Pei. Cleaning crowdsourced labels using oracles for statistical classification. *Proc. VLDB Endow.*, 12(4):376–389, 2018.

[14] H. R. Ehrenberg, J. Shin, A. J. Ratner, J. A. Fries, and C. Ré. Data programming with ddlite: putting humans in a different part of the loop. In *HILDA@SIGMOD 2016*, page 13.

[15] T. M. M. et.al. Never-ending learning. In *AAAI 2015*, pages 2302–2310.

[16] J. Fan, G. Li, B. C. Ooi, K. Tan, and J. Feng. icrowd: An adaptive crowdsourcing framework. In *SIGMOD, 2015*, pages 1015–1030.

[17] J. Fan, M. Lu, B. C. Ooi, W. Tan, and M. Zhang. A hybrid machine-crowdsourcing system for matching web tables. In *ICDE 2014*, pages 976–987.

[18] S. Faradani, B. Hartmann, and P. G. Ipeirotis. What's the right price? pricing tasks for finishing on time. In *AAAI Workshop 2011*.

[19] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. Crowddb: answering queries with crowdsourcing. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2011, Athens, Greece, June 12-16, 2011*, pages 61–72. ACM, 2011.

[20] R. Gomes, P. Welinder, A. Krause, and P. Perona. Crowdclustering. In *NIPS 2011*, pages 558–566.

[21] S. Guo, A. G. Parameswaran, and H. Garcia-Molina. So who won?: dynamic max discovery with the crowd. In *SIGMOD 2012*, pages 385–396.

[22] A. Y. Halevy, F. Korn, N. F. Noy, C. Olston, N. Polyzotis, S. Roy, and S. E. Whang. Goods: Organizing google's datasets. In *SIGMOD 2016*, pages 795–806.

[23] H. Heikinheimo and A. Ukkonen. The crowd-median algorithm. In B. Hartman and E. Horvitz, editors, *Proceedings of the First AAAI Conference on Human Computation and Crowdsourcing, HCOMP 2013, November 7-9, 2013, Palm Springs, CA, USA*. AAAI, 2013.

[24] C. Ho, S. Jabbari, and J. W. Vaughan. Adaptive task assignment for crowdsourced classification. In *ICML 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 534–542.

[25] M. Joglekar, H. Garcia-Molina, and A. G. Parameswaran. Evaluating the crowd with confidence. In *SIGKDD 2013*, pages 686–694.

[26] H. Kaplan, I. Lotosh, T. Milo, and S. Novgorodov. Answering planning queries with the crowd. *Proc. VLDB Endow.*, 6(9):697–708, 2013.

[27] D. R. Karger, S. Oh, and D. Shah. Iterative learning for reliable crowdsourcing systems. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. C. N. Pereira, and K. Q. Weinberger, editors, *NIPS 2011*, pages 1953–1961.

[28] H. Kim and Z. Ghahramani. Bayesian classifier combination. In N. D. Lawrence and M. A. Girolami, editors, *AISTATS 2012*, volume 22 of *JMLR Proceedings*, pages 619–627.

[29] S. Krishnan, M. J. Franklin, K. Goldberg, J. Wang, and E. Wu. Activeclean: An interactive data cleaning framework for modern machine learning. In *SIGMOD 2016*, pages 2117–2120.

[30] S. Krishnan, M. J. Franklin, K. Goldberg, and E. Wu. Boostclean: Automated error detection and repair for machine learning. *CoRR*, abs/1711.01299, 2017.

[31] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *SIGIR 1994*, pages 3–12.

[32] G. Li, J. Wang, Y. Zheng, and M. J. Franklin. Crowdsourced data management: A survey. In *33rd IEEE International Conference on Data Engineering, ICDE 2017, San Diego, CA, USA, April 19-22, 2017*, pages 39–40. IEEE Computer Society, 2017.

[33] K. Li, X. Zhang, and G. Li. A rating-ranking method for crowdsourced top-k computation. In *SIGMOD 2018*, pages 975–990.

[34] Q. Li, Y. Li, J. Gao, L. Su, B. Zhao, M. Demirbas, W. Fan, and J. Han. A confidence-aware approach for truth discovery on long-tail data. *Proc. VLDB Endow.*, 8(4):425–436, 2014.

[35] Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan, and J. Han. Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. In *SIGMOD 2014*, pages 1187–1198.

[36] Y. Li, F. Reiss, and L. Chiticariu. Systemt: A declarative information extraction system. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA - System Demonstrations*, pages 109–114. The Association for Computer Linguistics, 2011.

[37] Q. Liu, J. Peng, and A. T. Ihler. Variational inference for crowdsourcing. In *NIPS 2012*, pages 701–709.

[38] X. Liu, M. Lu, B. C. Ooi, Y. Shen, S. Wu, and M. Zhang. CDAS: A crowdsourcing data analytics system. *Proc. VLDB Endow.*, 5(10):1040–1051, 2012.

[39] F. Ma, Y. Li, Q. Li, M. Qiu, J. Gao, S. Zhi, L. Su, B. Zhao, H. Ji, and J. Han. Faitcrowd: Fine grained truth discovery for crowdsourced data aggregation. In *SIGKDD 2015*, pages 745–754.

[40] A. Marcus, D. R. Karger, S. Madden, R. Miller, and S. Oh. Counting with the crowd. *Proc. VLDB Endow.*, 6(2):109–120, 2012.

[41] M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relation extraction without labeled data. In *ACL 2009*, pages 1003–1011.

[42] B. Mozafari, P. Sarkar, M. J. Franklin, M. I. Jordan, and S. Madden. Scaling up crowd-sourcing to very large datasets: A case for active learning. *Proc. VLDB Endow.*, 8(2):125–136, 2014.

[43] N. Nakashole, M. Theobald, and G. Weikum. Scalable knowledge harvesting with high precision and high recall. In *WSDM 2011*, pages 227–236.

[44] A. G. Parameswaran, H. Park, H. Garcia-Molina, N. Polyzotis, and J. Widom. Deco: declarative crowdsourcing. In *CIKM 2012*, pages 1203–1212.

[45] A. G. Parameswaran, A. D. Sarma, H. Garcia-Molina, N. Polyzotis, and J. Widom. Human-assisted graph search: it's okay to ask questions. *Proc. VLDB Endow.*, 4(5):267–278, 2011.

[46] A. Ratner, S. H. Bach, H. R. Ehrenberg, J. A. Fries, S. Wu, and C. Ré. Snorkel: rapid training data creation with weak supervision. *VLDB J.*, 29(2-3):709–730, 2020.

[47] A. J. Ratner, C. D. Sa, S. Wu, D. Selsam, and C. Ré. Data programming: Creating large training sets, quickly. In *NIPS 2016*, pages 3567–3575.

[48] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy. Learning from crowds. *Journal of Machine Learning Research*, 11(4), 2010.

[49] F. Ricci, L. Rokach, and B. Shapira, editors. *Recommender Systems Handbook*. Springer, 2015.

[50] F. Rodrigues and F. C. Pereira. Deep learning from crowds. In S. A. McIlraith and K. Q. Weinberger, editors, *AAAI 2018*, pages 1611–1618.

[51] N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. In *(ICML 2001*, pages 441–448.

[52] A. D. Sarma, A. G. Parameswaran, H. Garcia-Molina, and A. Y. Halevy. Crowd-powered find algorithms. In *ICDE 2014*, pages 964–975.

[53] B. Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

[54] B. Settles and M. Craven. An analysis of active learning strategies for sequence labeling tasks. In *EMNLP 2008*, pages 1070–1079.

[55] B. Settles, M. Craven, and S. Ray. Multiple-instance active learning. In *NIPS 2007*, pages 1289–1296.

[56] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *COLT 1992*, pages 287–294.

[57] M. Venanzi, J. Guiver, G. Kazai, P. Kohli, and M. Shokouhi. Community-based bayesian aggregation models for crowdsourcing. In *WWW 2014*, pages 155–164.

[58] V. Verroios, P. Lofgren, and H. Garcia-Molina. tdp: An optimal-latency budget allocation strategy for crowdsourced MAXIMUM operations. In *SIGMOD 2015*, pages 1047–1062.

[59] J. Wang, T. Kraska, M. J. Franklin, and J. Feng. Crowder: Crowdsourcing entity resolution. *Proc. VLDB Endow.*, 5(11):1483–1494, 2012.

[60] J. Wang, S. Krishnan, M. J. Franklin, K. Goldberg, T. Kraska, and T. Milo. A sample-and-clean framework for fast and accurate query processing on dirty data. In *SIGMOD 2014*, pages 469–480.

[61] J. Wang, G. Li, T. Kraska, M. J. Franklin, and J. Feng. Leveraging transitive relations for crowdsourced joins. In *SIGMOD 2013*, pages 229–240.

[62] S. Wang, X. Xiao, and C. Lee. Crowd-based deduplication: An adaptive approach. In *SIGMOD 2015*, pages 1263–1277.

[63] P. Welinder, S. Branson, P. Perona, and S. J. Belongie. The multidimensional wisdom of crowds. In *Advances in neural information processing systems*, pages 2424–2432, 2010.

[64] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. R. Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *NIPS 2009*, pages 2035–2043.

[65] Z. Xu, R. Akella, and Y. Zhang. Incorporating diversity and density in active learning for relevance feedback. In *ECIR 2007*, volume 4425 of *Lecture Notes in Computer Science*, pages 246–257.

[66] M. Yakout, K. Ganjam, K. Chakrabarti, and S. Chaudhuri. Infogather: entity augmentation and attribute discovery by holistic matching with web tables. In *SIGMOD 2012*, pages 97–108.

[67] V. K. Yalavarthi, X. Ke, and A. Khan. Select your questions wisely: For entity resolution with crowd errors. In *CIKM 2017*, pages 317–326.

[68] T. Yan, V. Kumar, and D. Ganesan. Crowdsearch: exploiting crowds for accurate real-time image search on mobile phones. In *MobiSys 2010*, pages 77–90.

[69] C. Zhang, J. Shin, C. Ré, M. J. Cafarella, and F. Niu. Extracting databases from dark data with deepdive. In *SIGMOD 2016*, pages 847–859.

[70] C. J. Zhang, L. Chen, H. V. Jagadish, and C. C. Cao. Reducing uncertainty of schema matching via crowdsourcing. *Proc. VLDB Endow.*, 6(9):757–768, 2013.

[71] C. J. Zhang, Y. Tong, and L. Chen. Where to: Crowd-aided path selection. *Proc. VLDB Endow.*, 7(14):2005–2016, 2014.

[72] Z. Zhao, F. Wei, M. Zhou, W. Chen, and W. Ng. Crowd-selection query processing in crowdsourcing databases: A task-driven approach. In *EDBT 2015*, pages 397–408.

[73] Z. Zhao, D. Yan, W. Ng, and S. Gao. A transfer learning based framework of crowd-selection on twitter. In *KDD 2013*, pages 1514–1517.

[74] Y. Zheng, R. Cheng, S. Maniu, and L. Mo. On optimality of jury selection in crowdsourcing. In *EDBT 2015*, pages 193–204. OpenProceedings.org.

[75] Y. Zheng, G. Li, and R. Cheng. DOCS: domain-aware crowdsourcing system. *Proc. VLDB Endow.*, 10(4):361–372, 2016.

[76] J. Zhong, K. Tang, and Z. Zhou. Active learning from crowds with unsure option. In Q. Yang and M. J. Wooldridge, editors, *IJCAI 2015*, pages 1061–1068.

[77] D. Zhou, J. C. Platt, S. Basu, and Y. Mao. Learning from the wisdom of crowds by minimax entropy. In *NIPS 2012*, pages 2204–2212.

An ML-Powered Human Behavior Management System

Sihem Amer-Yahia*, Reynold Cheng+, Mohamed Bouadi*, Abdelouahab Chibah*,
Mohammadreza Esfandiari*, Jiangping Zhou+, Nan Zhang+, Eric Lau+, Yuguo Li+,
Xiaolin Han+, Shivansh Mittal+
*Univ. Grenoble Alpes, CNRS, France,
{firstname.lastname}@univ-grenoble-alpes.fr
+University of Hong Kong, ckcheng@cs.hku.hk,
{zhoujp, liyg, zhangnan, ehylau, xiaolinh, shivansh}@hku.hk

Abstract

Our work aims to develop novel technologies for building an efficient data infrastructure as a backbone for a human behavior management system. Our infrastructure aims at facilitating behavior modeling, discovery, and exploitation, leading to two major outcomes: a behavior data management back-end and a high-level behavior specification API that supports mining, indexing and search, and AI-powered algorithms that provide the ability to extract insights on human behavior and to leverage data to advance human capital. We discuss the role of ML in populating and maintaining the back-end, and in exploiting it for human interest.

1 Introduction

We make a case for building a human behavior management system, where human behavior is a first class citizen and is mined, queried and managed over time. While several efforts have focused on studying human behavior in large scale population studies, in mining customer purchase patterns, or on the social Web, there is no single architecture that provides the ability to mine, query and manage behavior. Such a system would encourage reproducibility and enable several applications that benefit humans. In particular, the ability to model individual and collective behavior enables to offer new functionalities that let everyone can share and discover all kinds of assets, and combine assets to advance human capital. Assets can be composed into learning strategies that everyone can use to propose their skills, acquire new skills, or enhance existing ones. The proposed research will contribute to designing and developing approaches that leverage ML for building an effective and efficient human behavior management back-end that represents the variety of behaviors and caters to human needs. To enable this work, two novel and challenging research axes need to be developed in parallel: a database system to manage human behavior and a set of ML-powered algorithms that populate and maintain that database, as well as approaches to search and leverage behavior and assets with the goal of studying human behavior and advancing their capital.

To design a human behavior database, we need to capture human factors and populate the database by mining behavioral patterns over time. We propose to leverage approaches that estimate human factors [56] and mine

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

behaviors at the individual and collective levels [34, 46, 49]. The highly dynamic nature of human factors and behavior renders the maintenance of such a database quite challenging. We propose to leverage ML approaches to (i) learn behavior change rates, (ii) develop adaptive approaches to cater to humans, and (iii) choose back-end maintenance strategies accordingly. Existing work that leverages ML for data management [62] is nascent and covers the use of ML for query optimization [40] or for database indexing [38]. Additionally, the ability to search and leverage behavior and assets with the goal of studying human behavior and advancing their capital requires the development of novel ML-powered algorithms.

To study human behavior, we plan to develop a flexible tool for the on-demand discovery of behavioral patterns and their exploration over time. Such functionality benefits a variety of stakeholders. It would enable the design of robust population studies, marketing strategies, and promotional campaigns. Data scientists, social scientists, Web application designers, marketers, and other domain experts, need a single destination to explore behavioral changes. To query human behavior, we will leverage work on querying time series. Systems like Qetch [47] and ShapeSearch [60] provide the ability to express powerful shape queries through sketches, natural language, and regular expressions with flexible time span and amplitude. However, those queries are solely shape-based and cannot be used to search for change induced by humans of interest. We will hence develop approaches to combine shape queries with querying humans and change intensity.

For leveraging assets to advance human capital, we will make use of recent work on virtual marketplaces [29]. In physical workplaces, learning strategies include scaffolding where assets are combined in alternating difficulty levels, and collaboration where humans learn from their interactions with higher-skilled peers [24, 37, 42]. In virtual marketplaces, a few studies focused on humans' ability to improve their skills by completing tasks [32], and how affinity between humans can be used to form teams that collaborate to produce high quality contributions while also improving skills [29]. Our observation is that optimizing for human capital should be seen as a multi-objective problem that accounts for several goals and quality control, cost reduction, and human effort. We will develop a framework to observe humans and leverage ML techniques to learn their abilities and adaptively suggest appropriate assets to them for advancing their capital while accounting for other goals.

2 Motivating examples

Example 1 (Mining and querying behavior.): Our first example motivates the need for sophisticated mining approaches to discover and model evolving human behavior. We consider two typical examples: mining customer behavior in retail and mining the behavior of citizens riding public transportation. Mining behavior can help identify purchase patterns in the first case, and specific rider groups in the second, e.g., familiar stranger groups who use public transportation during the same period. In both examples, evolving behavior captures changing habits such as customers reacting to promotional offers, or riders changing groups when in transit. The ability to query behavior changes is useful for people in charge to analyze usage trends. In the riders case, this would help them identify trends of daily travel populations over time and see their behaviors change when special events occur (e.g., during the COVID-19 crisis). Extracted insights must be made available through a powerful querying interface to instruct public policies (e.g., new COVID-19 measures). Additionally, they can be used to train behavior change models that will instruct data maintenance. By learning different change models, we will enable ML-powered back-end updates.

Example 2 (Assets for advancing capital.): Assets such as online courses or facts to be verified, can be used for advancing human capital and the system must help humans improve their skill and knowledge either individually or collaboratively (a.k.a. peer learning). Assume we have 3 humans: Mary, John and Sarah. Mining their behavior in an online course system would help determine their skill level (1 for Mary who is a novice, 3 for John who has an intermediate level, and 5 for Sarah who is an expert). Given the courses they consumed so far, our goal is to assign to Mary a batch of $k = 5$ assets that maximize her learning. Mary needs the support of John or Sarah to consume intermediate level assets. She cannot consume hard assets even with help. By exposing Mary to her peers' contributions, her learning potential is likely to increase [27]. Assigning over-challenging assets to her may result in frustration, and assigning under-challenging assets may lead to boredom.

In another example, we have a set of fact-checking tasks. Each task constitutes a collaborative asset, to be consumed by 12 individuals with varying skills. Each pair of individuals has an affinity that reflects how effectively they can collaborate based on their socio-demographics. Therefore, there are $\binom{12}{2}$ pairs of affinities forming a graph. One goal here is to divide the humans into 3 equi-sized groups of 4 members each so that peer learning is maximized.

3 Our system

3.1 Building a human behavior management database

The goal of this axis is to develop an integrated data model to represent human behavior, encompassing human factors and asset dimensions, and to leverage ML in maintaining the database. To make behavior and assets usable, we need to develop an API for populating, maintaining and accessing behavior and assets, including a declarative query language for expressing complex conditions on human factors and behavior and assets. Through these objectives it will be possible to offer querying human behavior and assets as a simple service promoting their reusability. Additionally, we will investigate performance aspects of this language and propose mechanisms for its efficient evaluation. The database needs to store raw human/asset data but also the results of extracting insights such as mining behavioral change of user groups. The evolving nature of human factors makes this particularly challenging. It is therefore necessary to apply ML approaches to learn behavioral change models and leverage them at maintenance time, when new raw data and new insights need to be stored in the back-end. This ML-powered approach will be compared against traditional batch and incremental maintenance approaches.

Objectives:

1. Design a model to capture, represent and manage human factors, human behavior and assets. Two kinds of factors must be considered: people-specific ones such as socio-demographic attributes, skill, reputation/trust, and motivation; and collaborative factors such as affinity and interaction models.
2. Design a model to capture and store extracted insights on individual and group behavior.
3. Investigate indexing mechanisms to retrieve and query behavioral change and assets efficiently.
4. Develop ML-powered algorithms for updating and managing evolving human behavior at individual and group levels.

3.2 Leveraging assets

This axis explores the study and querying of human behavior over time and the development of approaches for leveraging assets in different applications. In particular, we will focus on leveraging assets for advancing human capital. In our first endeavor, we will design queries that express behavior-aware change primitives. Our queries will be sent to the backend and need to be expressive enough to query behavior shapes and changes. This would require defining new scoring semantics that combine shape matching and intensity of change. Existing algorithms to match shape queries operate in time series and rely on splitting time using a fixed-size window and matching the query to each region in the window. We will leverage drift detection approaches on data streams [34] to handle time in a dynamic fashion.

In our second endeavor, we will study asset assignment, expecting that appropriate assignments will have a positive impact on the inherent learning capability of humans and on their overall performance. We focus on a common class of assets, "Knowledge assets" in Bloom's taxonomy of educational objectives [17, 39] such as image classification, text editing, labeling, fact checking, and speech transcription asset. A common problem we will tackle is: given a human h and a set of unconsumed assets, which sequence of k assets will maximize h's learning potential? Here, learning potential is the maximum possible improvement in h's skill.

We adopt a model where contributions from other humans are made visible to the current human. Several studies showed that humans learn better when contributions from higher-skilled humans are shown to them [25, 26, 35, 36]. Our challenges are: (1) how to choose an appropriate batch of k assets where a human can see others' contributions, (2) how to order the chosen k assets appropriately so that the human's skill improvement is maximized. Our approach must enabke both individual and peer learning. We will leverage work in online critiquing communities,[1] social Q&A sites,[2] and crowdsourcing platforms[3] that investigate how collaboration can promote knowledge and skill improvement of individuals. In particular, we propose to explore how affinity between group members improves peer learning and address modeling, theoretical, and algorithmic challenges. We will build on our recent work for algorithmic group formation with affinities for peer learning [29].

Objectives for querying behavior:

1. Formalize an algebra that captures behavior evolution over time.

2. Develop a framework that given raw human/asset data, extracts groups and their behavior, stores them as insights in the database and represents change using our algebra.

3. Build a visual interface to query behavior with powerful conditions on behavior shape and change intensity.

Objectives for learning:

1. Formalize the *learning potential* of a human for an asset and choose k assets that maximize the total learning potential. This formalization should capture individual learning and peer learning. There are two theories underlying our framework. First, Zone of Proximal Development (ZPD) [65] is a well-known theory that defines three zones of assets with different skill improvements; (1) A learnable zone that contains assets a person can learn how to consume when assisted by a teacher or peer with a higher skillset, (2) a flow/comfort zone of assets that are easy and can be consumed with no help, and (3) a frustration zone of assets that a learner cannot consume even with help. Second, the Flow theory [21] states that people are able to immerse themselves in doing things whose challenge matches their skills. In [15], the authors claim that to improve skills, the assets should be either in the flow/comfort zone, or in the learnable zone on the condition that there is some "scaffolding" to help humans consume assets that are a bit more challenging for them. This results in skill improvement (the dotted line). Our formalization builds on that and defines the learning potential for both individual assets (mainly in the flow/comfort zone) and collaborative assets (mainly in the learnable zone).

2. Devise learning strategies which interleave individual assets and collaborative assets. We will study their impact on humans' performance and skills. Previous work found that the order of assets impacts quality and completion time [22, 18]. For instance, assets could be provided in no particular order, or grouped and presented in alternating difficulty levels.

3. Propose adaptive and iterative asset search methods that take a human h, and assigns to h, at each iteration, a batch of k assets according to a learning objective. This approach may give rise to multi-objective problems.

[1] `https://movielens.org/`

[2] `http://quora.com/`

[3] `https://www.figure-eight.com/`

4 Modeling human behavior

Our model must capture humans, assets, and human behavior and its evolution over time. Figure 1 shows a two-level E/R diagram that represents human and asset data and extracted insights. It will serve as a basis for the design of our backend.

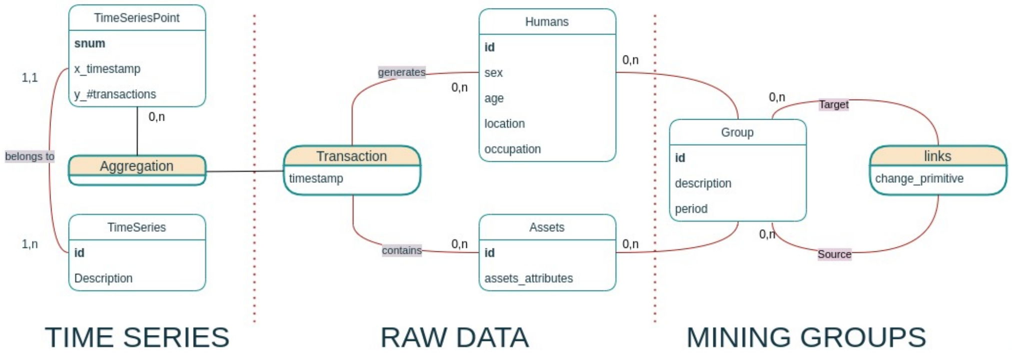


Figure 1: E/R diagram representing raw human/asset data and extracted insights.

The above framework could be used to discover and model evolving human behavior of citizens riding public transportation, and see how their behaviors change when special events occur. Here, we aim to study passenger behaviors during the COVID-19 crisis. COVID-19 is characterized by a broad spectrum of disease presentation, from asymptomatic or subclinical infections to severe diseases and deaths. A person infected with SARS-CoV-2 virus who has not yet developed symptoms or confirmed by laboratory testing would likely maintain normal social activities. Recent studies show that infected cases are contagious for asymptomatic and pre-symptomatic cases, and continuous to be contagious for at least a week [1, 2], providing ample opportunity for transmitting SARS-CoV-2 through public transportation. Recently, we have performed a study in Hong Kong with the Mass Transit Railway (MTR) Corporation, which is the only railway and subway service provider in Hong Kong. The MTR railway network covers areas inhabited by more than 70% of the local population [4]. About 50% of total number of rider trips in Hong Kong (or 4.5 million riders) are made through MTR [5]. Thus, the local population mobility in Hong Kong is well represented by the MTR traveling population. Courtesy of MTR, we have obtained *all* the entry and exit data of of anonymous riders (e.g., time and station of entry, ticket type (kid/adult/elder)) in the first four months (i.e., 1 January to 30 April) of 2020, which is also a coronavirus outbreak period in Hong Kong.

We have performed some initial analysis of the MTR data. Figure 2 shows the change of daily population in the first three months of 2020 (here, *octopus* refers to the most popular smart card in Hong Kong; *ticket* means the single entry ticket). We can see that the daily MTR traveling population is reduced by more than 40% after the end of Chinese New Year holiday on Jan 28, which can be related to the lockdown of Hubei, China on Jan 23, just before the holiday. Also, the MTR traveling population in weekends is more than 10% less than that in workdays. Another observation is that while the number of MTR passengers is generally decreasing, the number of new confirmed cases increases significantly.

Rider type discovery. The MTR riders could physically encounter one another, and temporally share different facilities or spaces such as stations, platforms, elevators, and carriages. They can concurrently share a crowded and small carriage for an extensive amount of time. We would also like to study different group behaviors of MTR riders in the period of January to March 2020, during which the outbreak of coronavirus occurs. As discussed in [67], these riders include:

- "Someone-like-you": they are groups of riders who share the same trip and stay close to each other, and simultaneously share trajectories for at least one trip.

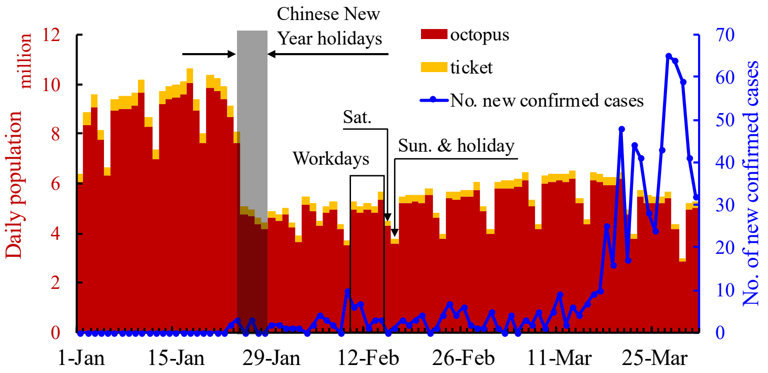


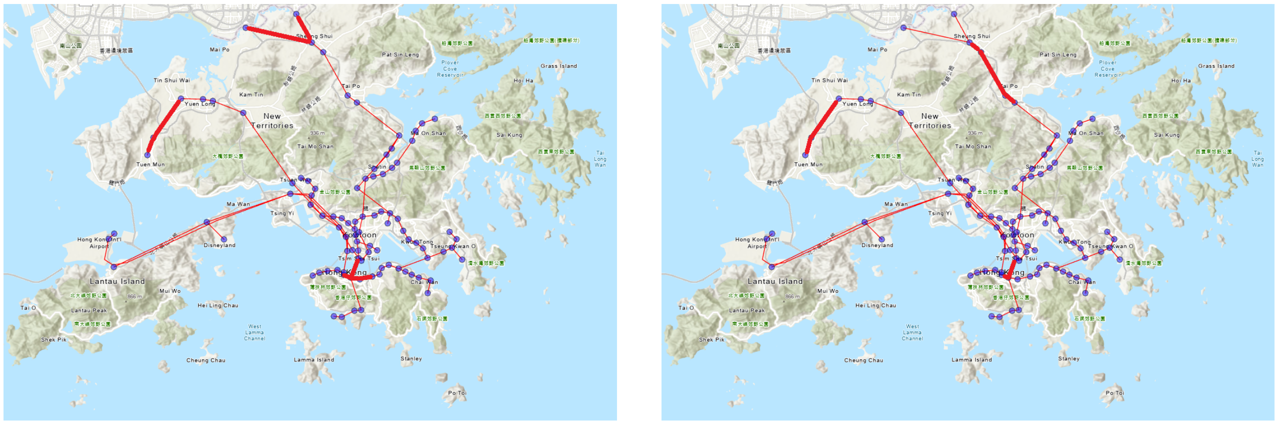
Figure 2: MTR daily population and number of infected cases in Jan-Mar 2020.

- Sensor riders, who have the most physical contacts with other riders at stations and carriages. If they are "super spreaders", they can infect many people.
- Extreme riders, who have to endure the longest journeys or who have to ride the MTR train most frequently. They can have a high risk of contracting coronavirus or respiratory diseases.
- Choice riders, who quickly change their travel patterns given the COVID-19 situation, for example, due to outbreak in certain districts, or new government policies for work and classes.

We aim to discover the above riders through mining the MTR data. The riders found can be useful to understand whether they have a high risk of spreading or contracting coronavirus. They can be useful for simulating scenarios and testing rider control and intervention policy. Moreover, they can be used for issuing warning or further actions. For example, in a "someone-like-you" group, if any member of this group contracts contagious viruses, other members who are close contacts of the patient should be notified and monitored as soon as possible. We will develop data mining solutions to identify rider types from the MTR data. For example, to obtain "someone-like-you" riders, one way is to group the rider information according to the stations and the times they entered and exited. Each group consists of riders who are estimated to take the same train. We will study methods for finding all these groups, and perform extensive analysis, including the estimation of average group size over weekdays and weekends, and examining the station-pairs that exhibits large group sizes. Figure 3 shows the spatial-temporal patterns of the average "someone-like-you" group sizes along different MTR trips during workdays in Hong Kong before Jan 24 and after Jan 28 in 2020. The thickness of the line indicates the size of the rider group. We can see that that the trips with the largest "someone-like-you" groups in the two time periods are not the same. We will compare the rider types across the first four months of 2019 and 2020.

Due to the gigantic number of riders, the discovery of rider types can be time-consuming. We will develop fast database algorithms and indexes to discover riders efficiently. We will extend the TopPI algorithm [44], developed by co-I Amer-Yahia, in order to mine rider groups from millions of records in sub-seconds. We will also extend the work on exploring datasets with rating maps to enable group comparison [11].

Our initial studies show that the data in hand is very promising (e.g., find out the risks of contracting coronavirus for different groups of passengers, and riders' travel behavior arising from the anxiety of being infected in public transport). However, without a scalable platform for human behavior analysis, it is difficult to perform advanced analysis and simulation. In particular, the MTR data, more than 60GB, contains more than 1 billion entry/exit transactions of 8 million riders. We would like to perform "microscopic" analysis (e.g., do riders have a close contact?), and also see how the disease is spread in a crowded train with insufficient ventilation. As the MTR data spans 4 months, it is also interesting to examine how the behaviors of passengers change with time. Existing practices and solutions in the public health and transportation fields are not scalable to handle such a



Workdays on and before Jan 24 **Workdays after Jan 28**

Figure 3: Spatio-temporal patterns of "someone-like-you" riders in MTR.

large amount of data (e.g., it takes more than 5 hours to perform a task to analyze rider behavior for a single day's data). This is simply too slow in face of the crisis that we are facing. The proposed system would be able to support analysis on the group behavior of citizens taking public transportation. Our results would help to monitor population transmission potential, and guide the appropriate level of social distancing measures [3].

5 Leveraging assets for learning

We consider a set of humans $\mathcal{H}$ and a set of assets $\mathcal{A}$. Humans in $\mathcal{H}$ consume assets in $\mathcal{A}$ at different times, either together or separately. The term asset is general enough to represent recommendations on the social web, courses in an online teaching system, tasks in crowdsourcing, etc. We denote $\mathbf{A}_h^t$ a batch of (possibly ordered) assets consumed by a human h at time t. The learning potential of a human h who consumes a batch of assets $\mathbf{A}$ at time $t+1$ depends on several factors: 1) the effort of h to consume assets in $\mathbf{A}$, 2) h's performance factor, 3) other humans' performance when learning collaboratively, and 4) the affinity between h and other humans. This gives rise to two problems: individual learning and collaborative learning.

We can define learning as the problem of individual asset assignment as follows: Given a human h and the batches of assets consumed by h up to iteration i: $\mathbf{A}_h^1 \ \ldots \ \mathbf{A}_h^t$, find a batch $\mathbf{A}$ of at most k assets to assign to human h at time $t+1$ such that:

$$\underset{\mathbf{A}}{\operatorname{argmax}}\, \mathit{learning}(h, \mathbf{A})$$

$\mathit{learning}(h, \mathbf{A})$ is a function that captures the learning potential of h who consumes the set of assets $\mathbf{A}$. Our problem is to determine the right batch of assets to provide to a human at time t. Our problem can be seen as a variant of the Knapsack Problem [19]. Our items are assets and each asset has a value (in our case v is $\mathit{learning}(w, t)$) and a weight, we want to find k assets that maximize the sum of values $\sum v_i$ under a capacity constraint k. What makes our problem simple is that the weight is equal to 1 which yields a top-k solution. Additionally, as the value of assigning an asset to a human depends on the human and evolves over time as other humans consume assets, we need to account for that dynamicity in the asset assignment process.

We can also define a variant of our problem where assets are consumed collaboratively: Given a set $H = \{h_1, ..., h_n\}$ of humans with their corresponding skill values h_t^s, our goal is to form a grouping $\mathcal{G}$ that contains k equi-sized groups $g_1, g_2, ..., g_k$ and that maximizes two objective functions, aggregated learning potential (*LP*) and aggregated affinity (*Aff*) between humans in the same group. More formally:

$$\underset{\mathcal{G}}{\text{maximize}} \quad \sum_{i=1}^{k} LP(g_i), \sum_{i=1}^{k} \mathit{Aff}(g_i) \;\text{ s.t. }\; |\mathcal{G}| = k,\; |g_i| = \frac{n}{k} \tag{1}$$

where $LP(g_i)$ (resp. $\mathit{Aff}(g_i)$) refers to any of the learning potential (resp. affinity).

Since the two objectives are incompatible with one another, our problem qualifies as *multi-objective*. In [29], we present approximation algorithms that find a feasible grouping (that maximizes learning potential) and offer provable constant approximation for affinities. We plan to build on that work.

Whenever a human consumes a collaborative asset, the asset's metadata is updated. Additionally, assets are grouped by difficulty level and by the number of remaining humans. A human's performance factor and skill also need to be updated as humans consume assets. We assume that a human's skill improves monotonically: the skill level remains the same or increases as time passes [48, 64] and is updated as they consume more assets. An ML approach that observes humans and revisits their attributes is warranted.

6 Related work

6.1 Mining customer behavior

Several approaches were proposed to track the evolution of customer groups over time [34, 46], and detect behavioral changes using pattern mining. Starting from a transactional database with customer demographics, the RFM score [49] is used to create customer groups according to their purchase frequency and spending. Association rules are extracted for consecutive time periods and compared with a custom similarity measure that leads to four changes: emerging, added, perished and unexpected patterns.

In [45], customer groups are built to reflect long-term patterns such as seasonal effects, and short-term patterns such as attractiveness of promotions. Customer purchases are captured with a non-homogeneous Poisson Process. The aim is to identify for a given product and period, k latent overlapping customer groups. Model parameters are learned with an Expectation–Maximization algorithm. Experiments on an Australian supermarket show that long-term and short term patterns provide insights such as ”If the demand for a product category is seasonal, the category will have more U-shape and inverse U-shape patterns”.

We plan to enable the application of different approaches for mining behavioral change. More importantly, the result of behavior mining will be stored in our back-end and readily available for querying.

6.2 Peer learning

Social science has a long history of studying non-computational aspects of computer-supported collaborative learning [20, 23]. With the development of online educational platforms (such as, Massive Open Online Courses or MOOCs), several parameters were identified for building effective teams: (1) individual and group learning and social goals, (2) interaction processes and feedbacks [61], (3) roles that determine the nature and group idiosyncrasy [23]. To the best of our knowledge, the closest to our work are [6, 7, 9], where quantitative models are proposed to promote group-based learning, albeit without affinity.

Our work is grounded in social science and takes a computational approach to the design of scalable solutions for peer learning with guarantees.

Many papers in crowdsourcing report that making other humans' contributions visible improves skills during task completion [27, 41, 35, 36, 26]. We will use this kind of indirect communication among humans in our collaborative assets. Group formation in online communities has been studied primarily in the context of task assignment [12, 13, 43, 16, 54]. The problem is often stated as: given a set of individuals and tasks, form a set of groups that optimize some aggregated utility subject to constraints such as group size, maximum workload etc. Utility can be aggregated in different ways: the sum of individual skills, their product, etc [13]. Group formation is combinatorial in nature and proposed algorithms solve the problem under different constraints and utility definitions (e.g., [43]).

Our work studies computational aspects and formulates optimization problems to find the best assets for a human. In particular, we leverage expressive multi-objective formulations to optimize more than one goal and form groups with the goal of maximizing peer learning under different affinities.

6.3 Querying change

Visual querying tools [50, 52, 57, 59, 66] help search for time series containing a desired shape by taking as input a sketch. Most of these tools perform precise point-wise matching using measures such as Euclidean distance or DTW. A few others enable flexible search and define a scoring function to capture how well a time series matches a sketch. Tools like TimeSearcher [14] let users apply soft or hard constraints on the x and y range values via boxes or query envelopes, but do not support other shape primitives beyond location constraints. Qetch [47] supports visual sketches and a custom similarity metric that is robust to distortions in the query, in addition to supporting a “repeat” operator for finding recurring patterns. ShapeSearch [60] enables expressive shape queries in the form of sketches, natural-language, and visual regular expressions. Queries are translated into a shape algebra and evaluated efficiently. Symbolic sequence matching papers approach the problem of pattern matching by employing offline computation to chunk trendlines into fixed length blocks, encoding each block with a symbol that describes the pattern in that block [10, 31, 33, 8, 58]. Among those, Shape Definition Language (SDL) [10] encodes search blocks using “up”, “down”, and “flat” patterns, much like Qetch and ShapeSearch, and supports a language for searching for patterns based on their sequence or the number of occurrences. A few other visual time series exploration tools such as Metro-Viz [28] and ONEX [53] support additional analytics tasks such as anomaly detection and clustering.

Our work is complementary to ShapeSearch and Qetch. We will extend Qetch with the ability to detect and score finer changes for a set of humans.

7 Conclusion

Emerging Big Data applications, such as learning the behaviors of citizens taking public transportation and providing them with assets for advancing human capital, necessitates the development of a scalable ML-driven human behavior management system. Such a system not only enables the development of human-behavior learning applications, but also provides insights on how to properly enable updates, which is important to these applications in which new data are generated at high speeds. An immediate direction is to study ML-driven data maintenance, which governs data update strategies based on machine learning approaches.

References

[1] He X, Lau EHY, Wu P, Deng X, Wang J, Hao X, Lau YC, Wong JY, Guan Y, Tan X, Mo X, Chen Y, Liao B, Chen W, Hu F, Zhang Q, Zhong M, Wu Y, Zhao L, Zhang F, Cowling BJ, Li F, Leung GM. Temporal dynamics in viral shedding and transmissibility of COVID-19. *Nat Med*. 2020 May;26(5):672-675.

[2] Emery JC, Russell TW, Liu Y, Hellewell J, Pearson CA; CMMID COVID-19 Working Group, Knight GM, Eggo RM, Kucharski AJ, Funk S, Flasche S, Houben RMGJ. The contribution of asymptomatic SARS-CoV-2 infections to transmission on the Diamond Princess cruise ship. *Elife*. 2020 Aug 24;9:e58699.

[3] Buckee CO, Balsari S, Chan J, et al. Aggregated mobility data could help fight COVID-19. *Science*. 2020;368(6487):145-146.

[4] Transport and Housing Bureau. Railway development strategy. *HKSAR Government*, 2014.

[5] Transport Department. Public transport strategy study. *HKSAR Government*, 2017.

[6] Rakesh Agrawal, Behzad Golshan, and Evangelos Papalexakis. Toward data-driven design of educational courses: A feasibility study. *EDM*, 2016.

[7] Rakesh Agrawal, Behzad Golshan, and Evimaria Terzi. Grouping students in educational settings. In *SIGKDD*, 2014.

[8] Rakesh Agrawal, King-Ip Lin, Harpreet S. Sawhney, and Kyuseok Shim. Fast similarity search in the presence of noise, scaling, and translation in time-series databases. In Umeshwar Dayal, Peter M. D. Gray, and Shojiro Nishio, editors, *VLDB'95, Proceedings of 21th International Conference on Very Large Data Bases, September 11-15, 1995, Zurich, Switzerland*, pages 490–501. Morgan Kaufmann, 1995.

[9] Rakesh Agrawal, Sharad Nandanwar, and Narasimha Murty Musti. Grouping students for maximizing learning from peers. In *EDM*, 2017.

[10] Rakesh Agrawal, Giuseppe Psaila, Edward L. Wimmers, and Mohamed Zaït. Querying shapes of histories. In Umeshwar Dayal, Peter M. D. Gray, and Shojiro Nishio, editors, *VLDB'95, Proceedings of 21th International Conference on Very Large Data Bases, September 11-15, 1995, Zurich, Switzerland*, pages 502–514. Morgan Kaufmann, 1995.

[11] Sihem Amer-Yahia, Sofia Kleisarchaki, Naresh Kumar Kolloju, Laks V. S. Lakshmanan, Ruben H. Zamar. Exploring Rated Datasets with Rating Maps. In Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, 2017, 1411–1419, 2017.

[12] Aris Anagnostopoulos et al. Power in unity: forming teams in large-scale community systems. In *CIKM*, 2010.

[13] Aris Anagnostopoulos et al. Online team formation in social networks. In *WWW*, 2012.

[14] A Aris, A Khella, P Buono, B Shneiderman, and C Plaisant. Timesearcher 2. *Human-Computer Interaction Laboratory, Computer Science Department, University of Maryland*, 2005.

[15] Ashok R Basawapatna, Alexander Repenning, Kyu Han Koh, and Hilarie Nickerson. The zones of proximal flow: guiding students through a space of computational thinking skills and challenges. In *Proceedings of the ninth annual international ACM conference on International computing education research*, pages 67–74, 2013.

[16] Senjuti Basu Roy et al. Task assignment optimization in knowledge-intensive crowdsourcing. *VLDA*, 2015.

[17] Benjamin S Bloom. Taxonomy of educational objectives. vol. 1: Cognitive domain. *New York: McKay*, pages 20–24, 1956.

[18] Carrie J Cai, Shamsi T Iqbal, and Jaime Teevan. Chain reactions: The impact of order on microtask chains. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 3143–3154. ACM, 2016.

[19] Chandra Chekuri and Sanjeev Khanna. A polynomial time approximation scheme for the multiple knapsack problem. *SIAM J. COMPUT*, 38(3):1, 2006.

[20] Elizabeth G Cohen. Restructuring the classroom: Conditions for productive small groups. *Review of educational research*, 1994.

[21] Mihaly Csikszentmihalyi. *Beyond boredom and anxiety: The experience of play in work and games.* Jossey-Bass, 1975.

[22] Peng Dai, Jeffrey M Rzeszotarski, Praveen Paritosh, and Ed H Chi. And now for something completely different: Improving crowdsourcing workflows with micro-diversions. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pages 628–638. ACM, 2015.

[23] Thanasis Daradoumis et al. Supporting the composition of effective virtual groups for collaborative learning. In *ICCE*. IEEE, 2002.

[24] Leo J De Vin, Lasse Jacobsson, JanErik Odhe, and Anders Wickberg. Lean production training for the manufacturing industry: Experiences from karlstad lean factory. *Procedia Manufacturing*, 11:1019–1026, 2017.

[25] Mira Dontcheva, Robert R Morris, Joel R Brandt, and Elizabeth M Gerber. Combining crowdsourcing and learning to improve engagement and performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3379–3388, 2014.

[26] Shayan Doroudi, Ece Kamar, and Emma Brunskill. Not everyone writes good examples but good examples can come from anywhere. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 7, pages 12–21, 2019.

[27] Steven Dow, Anand Kulkarni, Scott Klemmer, and Björn Hartmann. Shepherding the crowd yields better work. In *Proceedings of the ACM 2012 conference on computer supported cooperative work*, pages 1013–1022, 2012.

[28] Philipp Eichmann, Franco Solleza, Nesime Tatbul, and Stan Zdonik. Visual exploration of time series anomalies with metro-viz. In Peter A. Boncz, Stefan Manegold, Anastasia Ailamaki, Amol Deshpande, and Tim Kraska, editors, *Proceedings of the 2019 International Conference on Management of Data, SIGMOD Conference 2019, Amsterdam,*

The Netherlands, June 30 - July 5, 2019, pages 1901–1904. ACM, 2019.

[29] Mohammadreza Esfandiari, Dong Wei, Sihem Amer-Yahia, and Senjuti Basu Roy. Optimizing peer learning in online groups with affinities. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 1216–1226, 2019.

[30] Sarah Evans et al. More than peer production: Fanfiction communities as sites of distributed mentoring. In *CSCW*, 2017.

[31] Christos Faloutsos, M. Ranganathan, and Yannis Manolopoulos. Fast subsequence matching in time-series databases. In Richard T. Snodgrass and Marianne Winslett, editors, *Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data, Minneapolis, Minnesota, USA, May 24-27, 1994*, pages 419–429. ACM Press, 1994.

[32] Ujwal Gadiraju and Stefan Dietze. Improving learning through achievement priming in crowdsourced information finding microtasks. In *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*, pages 105–114. ACM, 2017.

[33] Minos N. Garofalakis, Rajeev Rastogi, and Kyuseok Shim. SPIRIT: sequential pattern mining with regular expression constraints. In Malcolm P. Atkinson, Maria E. Orlowska, Patrick Valduriez, Stanley B. Zdonik, and Michael L. Brodie, editors, *VLDB'99, Proceedings of 25th International Conference on Very Large Data Bases, September 7-10, 1999, Edinburgh, Scotland, UK*, pages 223–234. Morgan Kaufmann, 1999.

[34] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. Detecting change in data streams. In *(e)Proceedings of the Thirtieth International Conference on Very Large Data Bases, VLDA 2004, Toronto, Canada, August 31 - September 3 2004*, pages 180–191, 2004.

[35] Juho Kim. *Learnersourcing: improving learning with collective learner activity*. PhD thesis, Massachusetts Institute of Technology, 2015.

[36] Masaki Kobayashi, Hiromi Morita, Masaki Matsubara, Nobuyuki Shimizu, and Atsuyuki Morishima. An empirical study on short-and long-term effects of self-correction in crowdsourced microtasks. In *Sixth AAAI Conference on Human Computation and Crowdsourcing*, 2018.

[37] Martijn Koops and Martijn Hoevenaar. Conceptual change during a serious game: Using a lemniscate model to compare strategies in a physics game. *Simulation & Gaming*, 44(4):544–561, 2013.

[38] Tim Kraska, Alex Beutel, Ed H. Chi, Jeffrey Dean, and Neoklis Polyzotis. The case for learned index structures. In *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018*, pages 489–504, 2018.

[39] David R Krathwohl and Lorin W Anderson. *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives*. Longman, 2009.

[40] Ani Kristo, Kapil Vaidya, Ugur Çetintemel, Sanchit Misra, and Tim Kraska. The case for a learned sorting algorithm. In *Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14-19, 2020*, pages 1001–1016, 2020.

[41] Suna Kyun, Slava Kalyuga, and John Sweller. The effect of worked examples when learning to write essays in english literature. *The Journal of Experimental Education*, 81(3):385–408, 2013.

[42] Susanne P Lajoie and Alan Lesgold. Apprenticeship training in the workplace: Computer-coached practice environment as a new form of apprenticeship. *Machine-mediated learning*, 3(1):7–28, 1989.

[43] Theodoros Lappas, Kun Liu, and Evimaria Terzi. Finding a team of experts in social networks. In *SIGKDD*, 2009.

[44] Vincent Leroy, Martin Kirchgessner, Alexandre Termier, Sihem Amer-Yahia. TopPI: An efficient algorithm for item-centric mining. Inf. Syst. Volume 64, 104–118, 2017.

[45] Ling Luo, Bin Li, Irena Koprinska, Shlomo Berkovsky, and Fang Chen. Discovering temporal purchase patterns with different responses to promotions. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, page 2197–2202, New York, NY, USA, 2016. ACM.

[46] Ling Luo, Bin Li, Irena Koprinska, Shlomo Berkovsky, and Fang Chen. Tracking the evolution of customer purchase behavior segmentation via a fragmentation-coagulation process. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 2414–2420, 2017.

[47] Miro Mannino and Azza Abouzied. Qetch: Time series querying with expressive sketches. In Gautam Das, Christopher M. Jermaine, and Philip A. Bernstein, editors, *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018*, pages 1741–1744. ACM, 2018.

[48] Julian John McAuley and Jure Leskovec. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In *Proceedings of the 22nd international conference on World Wide WeA*, pages 897–908, 2013.

[49] R Miglausch John. Thoughts on rfm scoring. *The Journal of Database Marketing*, 8(1):7, 2000.

[50] Matt Mohebbi, Dan Vanderkam, Julia Kodysh, Rob Schonberger, Hyunyoung Choi, and Sanjiv Kumar. Google correlate whitepaper. 2011.

[51] Vicente Rodríguez Montequín et al. Using myers-briggs type indicator (MBTI) for assessment success of student groups in project based learning. In *CSEDU*, 2010.

[52] P. K. Muthumanickam, K. Vrotsou, M. Cooper, and J. Johansson. Shape grammar extraction for efficient query-by-sketch pattern matching in long time series. In *2016 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 121–130, 2016.

[53] Rodica Neamtu, Ramoza Ahsan, Charles Lovering, Cuong Nguyen, Elke A. Rundensteiner, and Gábor N. Sárközy. Interactive time series analytics powered by ONEX. In Semih Salihoglu, Wenchao Zhou, Rada Chirkova, Jun Yang, and Dan Suciu, editors, *Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD Conference 2017, Chicago, IL, USA, May 14-19, 2017*, pages 1595–1598. ACM, 2017.

[54] Habibur Rahman et al. Task assignment optimization in collaborative crowdsourcing. In *ICDM*, 2015.

[55] Habibur Rahman et al. Worker skill estimation in team-based tasks. *PVLDA*, 2015.

[56] Habibur Rahman, Saravanan Thirumuruganathan, Senjuti Basu Roy, Sihem Amer-Yahia, and Gautam Das. Worker skill estimation in team-based tasks. *Proceedings of the VLDB Endowment*, 8(11):1142–1153, 2015.

[57] Kathy Ryall, Neal Lesh, Tom Lanning, Darren Leigh, Hiroaki Miyashita, and Shigeru Makino. Querylines: Approximate query for visual browsing. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '05, page 1765–1768, New York, NY, USA, 2005. Association for Computing Machinery.

[58] Hagit Shatkay and Stanley B. Zdonik. Approximate queries and representations for large data sequences. *CoRR*, abs/1904.09262, 2019.

[59] Tarique Siddiqui, Albert Kim, John Lee, Karrie Karahalios, and Aditya G. Parameswaran. zenvisage: Effortless visual data exploration. *CoRR*, abs/1604.03583, 2016.

[60] Tarique Siddiqui, Paul Luh, Zesheng Wang, Karrie Karahalios, and Aditya G. Parameswaran. Shapesearch: A flexible and efficient system for shape-based exploration of trendlines. In David Maier, Rachel Pottinger, AnHai Doan, Wang-Chiew Tan, Abdussalam Alawini, and Hung Q. Ngo, editors, *Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14-19, 2020*, pages 51–65. ACM, 2020.

[61] Ivan Srba and Maria Bielikova. Dynamic group formation as an approach to collaborative learning support. *TLT*, 2015.

[62] Ion Stoica. Systems and ML: when the sum is greater than its parts. In *Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14-19, 2020*, page 1, 2020.

[63] Mohammad Taheri, Nasser Sherkat, Nick Shopland, Dorothea Tsatsou, Enrique Hortal Nicholas Vretos, Christos Athanasiadis, and Penny Standen. Adaptation and personalization principles based on mathisis findings. In *Public report on Managing Affective-learning THrough Intelligent atoms and Smart InteractionS project*, 2017.

[64] Kazutoshi Umemoto, Tova Milo, and Masaru Kitsuregawa. Toward recommendation for upskilling: Modeling skill improvement and item difficulty in action sequences. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pages 169–180. IEEE, 2020.

[65] Lev Vygotsky. Zone of proximal development. *Mind in society: The development of higher psychological processes*, 5291:157, 1987.

[66] Martin Wattenberg. Sketching a graph to query a time-series database. In Marilyn M. Tremaine, editor, *CHI '01 Extended Abstracts on Human Factors in Computing Systems, CHI Extended Abstracts '01, Seattle, Washington, USA, March 31 - April 5, 2001*, pages 381–382. ACM, 2001.

[67] J. Zhou, Y. Yang, H. Ma, and Y. Li. Familiar strangers in the big data era: An exploratory study of Beijing metro encounters. *Cities*, 97:102495, 2020.

Human-in-the-loop Artificial Intelligence for Fighting Online Misinformation: Challenges and Opportunities

Gianluca Demartini[1], Stefano Mizzaro[2], Damiano Spina[3]
[1]The University of Queensland, Brisbane, Australia, g.demartini@uq.edu.au
[2]University of Udine, Udine, Italy, mizzaro@uniud.it
[3]RMIT University, Melbourne, Australia, damiano.spina@rmit.edu.au

Abstract

The rise of online misinformation is posing a threat to the functioning of democratic processes. The ability to algorithmically spread false information through online social networks together with the data-driven ability to profile and micro-target individual users has made it possible to create customized false content that has the potential to influence decision making processes. Fortunately, similar data-driven and algorithmic methods can also be used to detect misinformation and to control its spread. Automatically estimating the reliability and trustworthiness of information is, however, a complex problem and it is today addressed by heavily relying on human experts known as fact-checkers. In this paper, we present the challenges and opportunities of combining automatic and manual fact-checking approaches to combat the spread on online misinformation also highlighting open research questions that the data engineering community should address.

1 Introduction

As the amount of online information that is generated every day in news, social media, and the Web increases exponentially, so does the harm that false, inaccurate, or incomplete information may cause to society. Experts in fact-checking organizations are getting overwhelmed by the amount of content that requires investigation,[1] and the sophistication of bots used to generate and deliberately spread fake news and false information (i.e., disinformation) is only making the tasks carried out by experts—i.e., identifying check-worthy claims and investigating the veracity of those statements—less manageable.

The aim of this paper is to discuss the main challenges and opportunities of a hybrid approach where Artificial Intelligence (AI) tools and humans—including both experts and non-experts recruited on crowdsourcing platforms—work together to combat the spread of online misinformation.

The remainder of this paper is organized as follows. Section 2 presents an overview of human-in-the-loop AI methods. Section 3 introduces the main challenges in identifying misinformation online. Section 4 summarizes recent work on machine learning methods applied to automatic truthfulness classification and check-worthiness. Section 5 describes recent advances on crowdsourcing one of the key activities in the fact-checking process, i.e.,

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

[1] `https://www.theverge.com/2020/3/24/21192206/snopes-coronavirus-covid-19-misinformation-fact-checking-staff`

judging the truthfulness or veracity of a given statement. Section 6 proposes a hybrid human-AI framework to fact-check information at scale. We conclude in Section 7 by summarizing the main take-away messages.

2 Human-in-the-loop AI

Human-in-the-loop AI (HAI) systems aim at leveraging the ability of AI to scale the processing to very large amounts of data while relying on human intelligence to perform very complex tasks— for example, natural language understanding—or to incorporate fairness and/or explainability properties into the system. Example of successful HAI methods include [8, 9, 15, 30]. Active learning methods [31] are another example of HAI where labels are collected from humans, fed back to a supervised learning model, and used to decide which data items humans should label next [32]. Related to this is the idea of interactive machine learning [2] where labels are automatically obtained from user interaction behaviors [20]. While being more powerful than pure machine-based AI methods, HAI systems need to deal with additional challenges to perform effectively and to produce valid results. One such challenge is the possible *noise* in the labels provided by humans. Depending on which human participants are providing labels for the AI component to learn from, the level of data quality may vary. For example, making use of crowdsourcing to collect human labels from people online either using paid micro-task platforms like Amazon Mechanical Turk or by means of alternative incentives like, e.g., 'games with a purpose' [37] is in general different from relying on a few experts.

There is often a trade-off between the cost and the quality of the collected labels. On the one hand, it may be possible to collect few high-quality curated labels that have been generated by domain experts, while, on the other hand, it may be possible to collect very large amounts of human-generated labels that might be not 100% accurate. Since the number of available experts is usually limited, to obtain both high volume and quality labels, the development of effective quality control mechanisms for crowdsourcing is needed.

Another challenge that comes with HAI systems is the *bias* that contributing humans may create and/or amplify in the annotated data and, consequently, in the models learned from this labelled data [16, 25]. Depending on the labelling task, bias and stereotypes of contributing individuals may be reflected into the generated labels. For example, an image labelling task that requires to identify the profession of people by looking at a picture, may lead to a female individual depicted in medical attire to be labelled as 'nurse' rather than as 'doctor'. For such type of data collection exercises, it becomes important to measure and, if necessary, control the bias in the collected data so that the bias in the AI models trained with such data is managed and controlled as well, if not limited or avoided altogether. Possible ways to control such bias include working on human annotator selection strategies by, for example, including pre-filtering tasks to profile annotators and to then select a balanced set of human annotators to generate labels for an AI to learn from.

Once manually labelled data has been collected, trained AI models may reflect existing bias in the data. An example of such a problem is that of 'unknown unknowns' (UU) [3], that is, data points for which a supervised model makes a high-confidence classification decision, which is however wrong. This means that the model is not aware of making mistakes. UUs are often difficult to identify because of the high-confidence of the model in its classification decision and may create critical issues in AI.[2] The problem of UU is usually caused by having a part of the feature space being under-represented in the training data (e.g., training data skewed towards white male images may result into AI models that are not performing well on images of people from other ethnicities and of other genders). Thus, such AI models are biased because of the unbalanced training data they have been trained on. Possible ways to control for such bias include making use of appropriate data sampling strategies to ensure that training datasets are well balanced and cover well the feature space also for features that may not have been originally identified or used.

[2] A classic example of this is the Google gorilla mistake, see https://www.wired.com/story/when-it-comes-to-gorillas-google-photos-remains-blind/.

When incorporating humans into an HAI system, they become the efficiency bottleneck. While purely machine-based AI systems can learn from very many data points and, once trained, perform decisions in real-time, making use of a human component makes the system less scalable and less efficient. For this reason, it becomes important to decide how to best employ these less efficient and limited human resources and, instead, how to best leverage the scalability of machine-based methods in order to get the best out of the two worlds. The problem becomes even more complex when considering different types of human contributors which come with varying quality, availability, and cost levels. We discuss this in more depth in Section 6.2.

Related to the previous problem, deciding what data points should be manually labelled by human annotators is another challenge. Given a usually very limited manual annotation budget, it becomes important to select the best data items to label in order to maximise their value with respect to the improvements of the trained AI models. Questions of this type are in particular relevant to systems relying on active learning strategies. Such improvements, however may relate not only to effectiveness, but also to other model properties like, for example, *fairness*. Another benefit of involving humans in HAI system is the ability to leverage their skills to improve the *interpretability* and *explainability* of AI models. Human contributors may be leveraged to, for example, add natural language explanations about *why* a certain supervised classifier decision has been made.

Thus, in order to design and develop an high-quality HAI system, researchers have to look at a multi-dimensional problem which includes aspects like efficiency, accuracy, interpretability, explainability, and fairness. Human and machine components of an HAI system can contribute and possibly threaten each of these dimensions. Based on these issues, the overarching question in HAI systems is about deciding *what should humans do* and *what should AIs do* in order to optimally leverage the capabilities of both methodologies. In the remainder of this paper we discuss these challenges and opportunities in the context of fighting online misinformation. We use this problem as a showcase of HAI methods and discuss the potential of such methodology when applied to this context.

3 The Problem of Online Misinformation

3.1 An Interdisciplinary Challenge

The spread of misinformation online is a threat to our safety online and risks to damage the democratic process. For instance, bots and trolls have been spreading disinformation across a number of significant scenarios, such as the election of US President Donald Trump in 2016 [5], the debate in the UK over Brexit [19], and, more recently, exaggerating the role of arson to undermine the link between bushfires in Australia and climate change.[3] The World Health Organization (WHO) has referred to the problem of large amount of misinformation spreading during the COVID-19 pandemic as an “infodemic”[4] [1]. Therefore, fact-checking information online is of great importance to avoid further costs to society.

Because of the importance, impact, and interdisciplinarity of the issue, a number of different research areas have focused on understanding and stopping misinformation spreading online. This includes research in political sciences [22], communication science [40], computational social science [7], up to computer science including the fields of human-computer interaction [33], database [21], and information retrieval [28]. While different research methodologies are being applied, the overarching goal is to understand how misinformation is spreading, why people trust it, and how to design and test systems and processes to stop it.

From a data engineering point of view, online misinformation poses some of the same common challenges observed in modern data management: i) *volume*: large amounts of data to be processed efficiently and in a scalable fashion; ii) *velocity*: processing data and making misinformation classification decisions in a timely

[3] `https://theconversation.com/bushfires-bots-and-arson-claims-australia-flung-in-the-global-disinformation-spotlight-129556`

[4] `https://www.who.int/dg/speeches/detail/director-general-s-remarks-at-the-media-briefing-on-2019-novel-coronavirus---8-february-2020`

fashion also in conditions when data to be checked comes as a stream (e.g., Twitter propaganda bots generating and propagating misinformation in social networks; iii) *variety*: misinformation comes in multiple formats, from textual statements in news articles, to images used in social media advertising, to deep-fake videos artificially generated by AI models; iv) *veracity*: the core question of truthfulness classification often translates in deciding which data source can be trusted and which not. Thus, the data engineering community not being new to dealing with such challenges, can surely provide solutions, systems, and tools able to support the fight to online misinformation. We however still believe that this is an interdisciplinary challenge, and in the remainder of this paper we present a framework that goes beyond data engineering by including humans in the loop and by considering human factors as well.

3.2 Misinformation Tasks

From the existing scientific literature about misinformation, we can see that there are a number of more specific tasks that need to be addressed to achieve the overarching goal of fighting online misinformation. The first task that comes to mind is *truthfulness classification*, that is, given a statement decide its truth level, in a scale from completely true to completely false. Fully automated approaches [23] as well as crowdsourcing-based approaches [28] have been proposed to address this task. However, other tasks related to online misinformation exist. For example, it is also important to decide about the *check-worthiness* of online content. As there are way too many statements and claims that could possibly be fact-checked, before expert fact checking can take place, a pre-processing filtering step needs to be completed to identify which statements should be going through a complete fact-checking process, out of a large collection of potential candidates. Criteria to be considered for such a selection process include: the potential harm that a certain statement being false could create, the reach of that statement, the importance and relevance of the topic addressed by the statement, etc. Automated methods for check-worthiness have been proposed in the literature [11], but are far from being effective enough to be deployed in practice and replace expert fact-checkers on this task.[5] Another task related to misinformation is *source identification*. Being able to detect the origin of online information can provide additional evidence to information consumers about its level of trustworthiness. More than just either manual or automatic approaches to address these tasks, an additional way is to combine them together in order to optimize processes and leverage the best properties of each method.

4 Machine Learning for Fighting Online Misinformation

For each of the misinformation tasks described in the previous section, there have been attempts to develop machine learning methods to tackle them. In this section we provide a summary of such research. For the problem of truthfulness classification, benchmarks on which to compare the effectiveness of different approaches have been developed. A popular benchmark for truthfulness classification is the LIAR dataset [39] that makes use of expert fact-checked statements from the PolitiFact website. More than 12K expert-labeled statements are used as ground truth to train and evaluate automatic classification systems effectiveness, so that system quality can be compared. Even larger than that is the FEVER dataset [34] that contains 180K statements obtained by altering sentences extracted from Wikipedia. Other earlier and smaller truthfulness classification benchmark datasets include [36, 14].

A lot of effort has been made within the AI research community not only to obtain accurate classification decision, but also to provide explainable results. Supervised methods for this task have looked at which features are the most indicative of truthfulness [27]. Recent approaches have designed neural networks that aim at combining evidence fragments together to inform the truthfulness classification decision [43]. Such evidence

[5] `https://www.niemanlab.org/2020/07/a-lesson-in-automated-journalism-bring-back-the-humans/`

can then be used to explain the automatic classification decisions. Other studies looking at the explainability dimension of this problem have observed that different features may be indicators for different types of fake news and can be used to cover different areas of the feature space [26]. Adversarial neural networks have shown to improve the effectiveness in identifying distinctive features for truthfulness classification [42].

Methods to automatically decide on check-worthiness [11] have looked at how to assign a score to a sentence and to predict the need for it to be checked by experts using supervised methods and training data. While some methods make use of contextual information, that is, of the surrounding text, to decide on the check-worthiness of a sentence [13], the most effective ones consider each sentence in isolation and use domain specific word embeddings within an LSTM network [17].

Metadata about information sources presented to social media users have an effect on the perceived truthfulness of the information [24]. Providing news source and contextual metadata may help users to make informed decisions [12]. Related to this, the New York Times R&D group has started a project to provide provenance metadata around news using blockchain technology to track the spread of news online and to provide contextual information to news readers.[6]

5 Crowdsourcing Truthfulness

More than just machine learning-based methods, crowdsourcing can be used as a way to label data at scale. In the context of misinformation, crowdsourcing is a methodology that can provide, for example, truthfulness classification labels for statements to be fact-checked. While experts may not be directly replaced by crowd workers (see work by Bailey et al. [4]), by deploying appropriate quality control mechanisms, crowdsourcing can provide reliable labels [10]. In a recent research on crowdsourcing truthfulness classification decisions we have looked at how to scale the collections of manual labels and at the impact of the annotators' background on the quality of the collected labels specifically looking for the impact of the annotator political bias with respect to the assessed statement and of the scale used to express the truthfulness judgment [28]. In another follow-up study, we have then looked at the impact of the *timeliness* of the assessed statements on the quality of the collected truthfulness labels. Results show that even more recent statements can still reliably be fact-checked by the crowd [29]. More in detail, we looked at how the crowd assessed the truthfulness of COVID-19 true and false statements during the pandemic, finding an agreement with expert judgments comparable to that in the previous study.

Another common challenge for expert fact-checkers, due to the limited available resources, is deciding which items should be fact-checked among very many candidates. More than just leveraging crowdsourcing to decide on truthfulness, the crowd may also be able to support expert fact-checkers in performing the task of deciding about the 'check-worthiness' of content, that is, asking the crowd to decide whether or not a given piece of content would benefit from being fact-checked by experts. Several factors affect the decision of selecting a statement to undergo a fact-checking process. The crowd may be involved in validating these factors which include, for example, the level of public interest of the assessed content, the possible impact of such content not being true, and the timeliness of the content. In this way, it would be possible to manually filter more content for fact-checking (the effectiveness of fully automated check-worthiness approach is still very low [11]) thus allowing expert fact-checkers to focus on actual fact-checking rather than on filtering and deciding what needs to be fact-checked.

[6] https://open.nytimes.com/introducing-the-news-provenance-project-723dbaf07c44

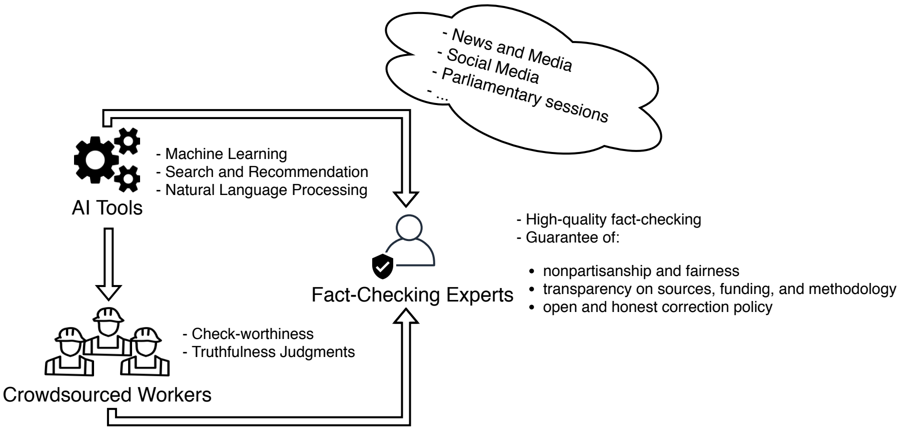

- News and Media
- Social Media
- Parliamentary sessions
- ...
AI Tools
- Machine Learning
- Search and Recommendation
- Natural Language Processing
Fact-Checking Experts
- High-quality fact-checking
- Guarantee of:
• nonpartisanship and fairness
• transparency on sources, funding, and methodology
• open and honest correction policy
Crowdsourced Workers
- Check-worthiness
- Truthfulness Judgments

Figure 1: Human-in-the-loop AI framework for fighting online misinformation.

6 A Hybrid Human-AI Framework for Fighting Online Misinformation

6.1 Combining Experts, AI, and Crowd

Given the limitations of both automated and human-based methods for fact checking, we rather envision a hybrid human-AI approach to fight online misinformation. Such an approach has the benefit of leveraging the positive aspects of each of the different approaches, that is, the scalability of AI to efficiently process very large amounts of data, the ability of expert fact-checkers to correctly identify the truthfulness level of verified statements in a transparent and fair way, and the ability of crowdsourcing to manually process significantly large datasets. We are starting to see the appearance of hybrid approaches for fact-checking, like, for example, the work presented by Karagiannis et al. [21]. The proposed system is an example of how to efficiently use human fact checking resources by having a machine-based system supporting them to find the facts that need to be manually checked out of a large database of possible candidates.

The combination of these methods may not only result in more efficient and effective fact-checking processes, but also lead to improved trust on the outcomes over purely AI-based methods and may also leverage the embedded human dimension to increase the level of transparency of the truthfulness labels attached to news (i.e., explaining *why* a certain piece of news has been labelled as fake, like fact-checkers do already, but something that AI-based methods still struggle to provide). Such an approach may also lead to resource optimization, where the more expensive and accurate expert fact checkers may be intelligently deployed only on the few most important and challenging verification tasks, while the crowd and AI can work together to scale-up the execution of very large amounts of fact-checking tasks. We thus envision a waterfall model where different levels of cost/quality trade-offs can be applied at different stages by means of appropriate task allocation models.

6.2 The Framework

The existence of numerous challenges and constraints that need to be resolved concurrently leads us to the proposal of a solution that not only combines humans and machines, but that in doing so leverages different types and levels of engagement in the process of fighting misinformation. Our proposed framework consists of three main actors: fact-checking experts, AI methods, and crowdsourcing workers (see Figure 1).

Fact-checking experts are the protagonists of the framework and are the ones who make use of the other two components to optimize the efficiency of the fact-checking process and maintain high-quality standards. Also,

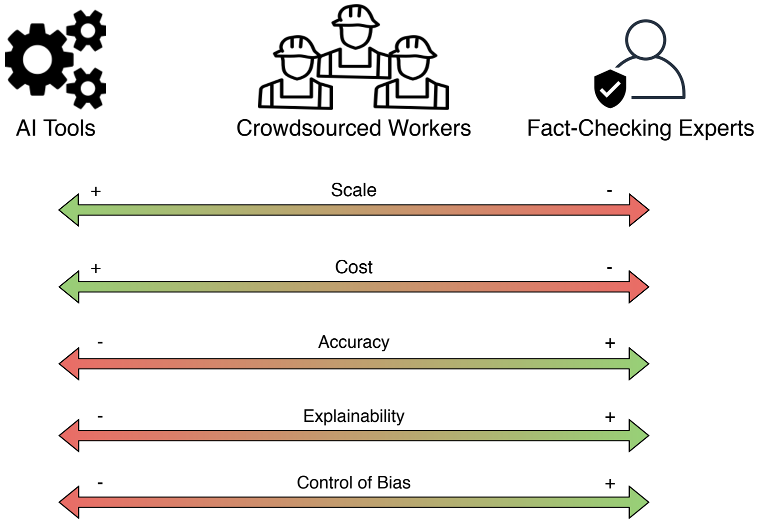


Figure 2: Trade-offs between the actors of the framework.

they are the only ones who can guarantee that this HAI system meets the three principles[7] of (i) non-partisanship and fairness, (ii) transparency on sources, funding, and methodology; and (iii) open and honest correction policy.

AI tools consist of automatic methods that fact-checkers can use to deal with the large amount of (mis)information produced through different channels such as news and media, parliamentary sessions, or social media [6]. Although AI tools are able to process data at scale, automatic predictions are typically not free from errors. For instance, machine learning methods used in systems such as ClaimBuster [18] or check-worthiness systems for the CheckThat! Lab at CLEF [11] are far from being 100% accurate. Moreover, it is not clear whether these tools would perform at the same level of accuracy in other scenarios, e.g., predicting check-worthiness of statements related to non-American politics. In summary, although state-of-the-art machine learning can compete—and even surpass—experts when data scale and costs are measured, as of today they are far from reaching human experts when considering the level of accuracy, explainability, and fairness.

Crowd workers somehow lie in between experts and AI on all the five above mentioned dimensions (scale, cost, accuracy, explainability, and control of bias) and can be deployed on-demand based on changing requirements and trade-offs. Figure 2 summarizes the strong and weak points of the actors involved in the proposed framework.

The proposed framework comes with several benefits:

- **Cost-quality trade-offs**: it comes with the ability to trade-off and optimize between required cost and quality of the label collection process where human experts (i.e., fact checkers) come with the highest quality and cost and AI comes with the lowest cost;
- **Load management**: it allows to deal with peaks of fact-checking tasks that may be otherwise impossible to deal with for expert fact-checkers working under constrained resource conditions. In such situations, they may be able to leverage the more scalable crowd and AI tools to deal with a sudden increase in annotation workload;
- **Trustworthiness**: it can serve as a way to make AI technology accepted in well-established traditional journalistic environments that would not see positively an ‘AI taking over their job’.

In such an intertwined framework, the key question becomes *who should do what*. Given a workload of misinformation tasks, a deadline, and required constraints like a minimum level of quality and a maximum

[7] https://ifcncodeofprinciples.poynter.org/know-more/the-commitments-of-the-code-of-principles

cost, the problem becomes to identify a task allocation solution that satisfies the constraints with maximum value. This can be addressed with a cascade model [35, 41, 38] with humans-in-the-loop, where AI tools, crowd workers and fact-checking experts cooperate to maximize value. For example, looking at the trade-off between *urgency* and quality, as soon as a statement is identified as requiring fact-check, an AI model can first be adopted to very efficiently provide a truthfulness label which could then possibly be replaced later on once a team of expert fact-checkers has concluded their forensic investigation of the available evidence in favour or against the statement being true. Such *cascade of annotation tasks* where many (or all) labels are quickly estimated automatically, only a small subset of those is sent to the crowd for a quick (but non-real time) validation of their truthfulness, and then only very few remaining statements are sent to experts to investigate in depth is the core idea of the proposed framework that leverages different levels of the size-quality-cost trade-offs that the different methodologies provide.

One dimension that impacts task allocation decisions is the cost and scale of the annotation problem. In order to leverage the best of the automated and manual methods, AI and crowdsourcing can be used to scale up the annotation effort to very many statements thus being able to possibly provide truthfulness labels for every single statement being published online. Expert fact-checkers can then be parsimoniously deployed on statements that are either difficult to label by AI or crowdsourcing methods (e.g., selected by means of low algorithmic confidence or low annotator agreement within the crowd), or important to label accurately due to the possibly wide implications of the statement being false or due to the importance of the speaker who made the statement and its potential reach.

Another open research question is on understanding how experts would actually work when embedded in this new framework: they would need to change consolidated and validated fact-checking processes and, instead, adapt to an environment in which their work is being complemented by AI and non-experts. This would necessarily require a certain level of trust in the HAI system that, on its side, is making decisions on what expert fact-checkers should do and on which statements they should work on. This translates into experts giving up a certain level of control on the process to the HAI system that has to decide what they do not get access to. For this to work, there needs to be a certain level of trust in the system that could possibly be achieved by the employment of self-explainable AI tools. This is also critical as as the fact-checking experts need at the end to be able to guarantee transparency on the process and methods used for fact-checking.

7 Take-Away Messages

In this paper we discussed the problem of online misinformation and proposed a hybrid human-AI approach to address it. We proposed a framework that combines AI, crowdsourcing, and expert fact-checkers to produce annotations for statements by balancing annotation cost, quality, volume, and speed thus providing information consumers (e.g., social media users) with timely and accurate fact-checking results at scale.

The proposed HAI approach aims at combining different methods to leverage the best properties of both AI and human-based annotation. Moreover, involving humans in the loop allows to better deal with the interdisciplinary nature of the misinformation problem by also providing human support on issues like explainability, trust, and bias.

The model presented in this paper envisions a complex collaborative scheme between different humans and different AIs where the open research question moves to the optimization of these complementary resources and on how to decide which task should be allocated to which element of the HAI system. A human-in-the-loop solution to misinformation can also provide increased transparency on fact-checking processes leveraging together algorithms and AI and, in the end, provide more evidence and power to the end users to make informed decisions on which online information they should and which they should not trust.

References

[1] F. Alam, S. Shaar, A. Nikolov, H. Mubarak, G. D. S. Martino, A. Abdelali, F. Dalvi, N. Durrani, H. Sajjad, K. Darwish, et al. Fighting the COVID-19 Infodemic: Modeling the Perspective of Journalists, Fact-Checkers, Social Media Platforms, Policy Makers, and the Society. *arXiv preprint arXiv:2005.00033*, 2020.

[2] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza. Power to the people: The role of humans in interactive machine learning. *AI Magazine*, 35(4):105–120, 2014.

[3] J. Attenberg, P. Ipeirotis, and F. Provost. Beat the machine: Challenging humans to find a predictive model's "unknown unknowns". *Journal of Data and Information Quality (JDIQ)*, 6(1):1–17, 2015.

[4] P. Bailey, N. Craswell, I. Soboroff, P. Thomas, A. P. de Vries, and E. Yilmaz. Relevance assessment: Are judges exchangeable and does it matter? In *Proceedings of SIGIR*, pages 667–674, 2008.

[5] A. Bovet and H. A. Makse. Influence of fake news in Twitter during the 2016 US presidential election. *Nature communications*, 10(1):1–14, 2019.

[6] A. Cerone, E. Naghizade, F. Scholer, D. Mallal, R. Skelton, and D. Spina. Watch 'n' Check: Towards a social media monitoring tool to assist fact-checking experts. In *Proceedings of DSAA*, 2020.

[7] M. Del Vicario, A. Bessi, F. Zollo, F. Petroni, A. Scala, G. Caldarelli, H. E. Stanley, and W. Quattrociocchi. The spreading of misinformation online. *PNAS*, 113(3):554–559, 2016.

[8] G. Demartini, D. E. Difallah, and P. Cudré-Mauroux. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st international conference on World Wide Web*, pages 469–478, 2012.

[9] G. Demartini, B. Trushkowsky, T. Kraska, M. J. Franklin, and U. Berkeley. Crowdq: Crowdsourced query understanding. In *CIDR*, 2013.

[10] G. Demartini, D. E. Difallah, U. Gadiraju, and M. Catasta. An introduction to hybrid human-machine information systems. *Foundations and Trends in Web Science*, 7(1):1–87, 2017.

[11] T. Elsayed, P. Nakov, A. Barrón-Cedeno, M. Hasanain, R. Suwaileh, G. Da San Martino, and P. Atanasova. Overview of the CLEF-2019 CheckThat! Lab: Automatic identification and verification of claims. In *Proceedings of CLEF*, pages 301–321, 2019.

[12] N. Evans, D. Edge, J. Larson, and C. White. News provenance: Revealing news text reuse at web-scale in an augmented news search experience. In *Proceedings of CHI*, pages 1–8, 2020.

[13] L. Favano, M. J. Carman, and P. L. Lanzi. TheEarthIsFlat's submission to CLEF'19 CheckThat! challenge. In *CLEF (Working Notes)*, 2019.

[14] W. Ferreira and A. Vlachos. Emergent: a novel data-set for stance classification. In *Proceedings of NAACL-HLT*, pages 1163–1168, 2016.

[15] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. Crowddb: answering queries with crowdsourcing. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 61–72, 2011.

[16] S. Hajian, F. Bonchi, and C. Castillo. Algorithmic bias: From discrimination discovery to fairness-aware data mining. In *Proceedings of KDD*, pages 2125–2126, 2016.

[17] C. Hansen, C. Hansen, S. Alstrup, J. Grue Simonsen, and C. Lioma. Neural check-worthiness ranking with weak supervision: Finding sentences for fact-checking. In *Proceedings of TheWebConf*, pages 994–1000, 2019.

[18] N. Hassan, G. Zhang, F. Arslan, J. Caraballo, D. Jimenez, S. Gawsane, S. Hasan, M. Joseph, A. Kulkarni, A. K. Nayak, et al. ClaimBuster: The first-ever end-to-end fact-checking system. *Proceedings of the VLDB Endowment*, 10(12): 1945–1948, 2017.

[19] P. N. Howard and B. Kollanyi. Bots, #StrongerIn, and #Brexit: Computational propaganda during the UK-EU referendum. *Available at SSRN 2798311*, 2016.

[20] T. Joachims and F. Radlinski. Search engines that learn from implicit feedback. *Computer*, 40(8):34–40, 2007.

[21] G. Karagiannis, M. Saeed, P. Papotti, and I. Trummer. Scrutinizer: A mixed-initiative approach to large-scale, data-driven claim verification. *Proceedings of the VLDB Endowment*, 13(11):2508–2521, 2020.

[22] J. H. Kuklinski, P. J. Quirk, J. Jerit, D. Schwieder, and R. F. Rich. Misinformation and the currency of democratic citizenship. *Journal of Politics*, 62(3):790–816, 2000.

[23] S. Miranda, D. Nogueira, A. Mendes, A. Vlachos, A. Secker, R. Garrett, J. Mitchel, and Z. Marinho. Automated fact checking in the news room. In *Proceedings of TheWebConf*, pages 3579–3583, 2019.

[24] A. Oeldorf-Hirsch and C. L. DeVoss. Who posted that story? processing layered sources in facebook news posts.

Journalism & Mass Communication Quarterly, 97(1):141–160, 2020.

[25] A. Olteanu, C. Castillo, F. Diaz, and E. Kiciman. Social data: Biases, methodological pitfalls, and ethical boundaries. *Frontiers in Big Data*, 2:13, 2019.

[26] J. C. Reis, A. Correia, F. Murai, A. Veloso, and F. Benevenuto. Explainable machine learning for fake news detection. In *Proceedings of WebSci*, pages 17–26, 2019.

[27] J. C. Reis, A. Correia, F. Murai, A. Veloso, and F. Benevenuto. Supervised learning for fake news detection. *IEEE Intelligent Systems*, 34(2):76–81, 2019.

[28] K. Roitero, M. Soprano, S. Fan, D. Spina, S. Mizzaro, and G. Demartini. Can the crowd identify misinformation objectively? The effects of judgment scale and assessor's background. In *Proceedings of SIGIR*, pages 439–448, 2020.

[29] K. Roitero, M. Soprano, B. Portelli, D. Spina, V. Della Mea, G. Serra, S. Mizzaro, and G. Demartini. The covid-19 infodemic: Can the crowd judge recent misinformation objectively? In *Proceedings of CIKM*, 2020. In press. arXiv preprint arXiv:2008.05701.

[30] C. Sarasua, E. Simperl, and N. F. Noy. Crowdmap: Crowdsourcing ontology alignment with microtasks. In *International semantic web conference*, pages 525–541. Springer, 2012.

[31] B. Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

[32] D. Spina, M.-H. Peetz, and M. de Rijke. Active learning for entity filtering in microblog streams. In *Proceedings of SIGIR*, pages 975–978, 2015.

[33] K. Starbird, A. Arif, and T. Wilson. Disinformation as collaborative work: Surfacing the participatory nature of strategic information operations. *PACMHCI*, 3(CSCW):1–26, 2019.

[34] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal. FEVER: A large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*, 2018.

[35] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of CVPR*, volume 1, pages I–I. IEEE, 2001.

[36] A. Vlachos and S. Riedel. Fact checking: Task definition and dataset construction. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pages 18–22, 2014.

[37] L. Von Ahn. Games with a purpose. *Computer*, 39(6):92–94, 2006.

[38] L. Wang, J. Lin, and D. Metzler. A cascade ranking model for efficient ranked retrieval. In *Proceedings of SIGIR*, pages 105–114, 2011.

[39] W. Y. Wang. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of ACL*, pages 422–426, 2017.

[40] C. Wardle. Fake news. It's complicated. *First Draft*, 16, 2017.

[41] D. Weiss and B. Taskar. Structured prediction cascades. In *Proceedings of AISTATS*, pages 916–923, 2010.

[42] L. Wu, Y. Rao, A. Nazir, and H. Jin. Discovering differential features: Adversarial learning for information credibility evaluation. *Information Sciences*, 516:453–473, 2020.

[43] L. Wu, Y. Rao, X. Yang, W. Wang, and A. Nazir. Evidence-aware hierarchical interactive attention networks for explainable claim verification. In *Proceedings of IJCAI*, pages 1388–1394, 2020.

Data Engineering

It's FREE to join!

TCDE

tab.computer.org/tcde/

The Technical Committee on Data Engineering (TCDE) of the IEEE Computer Society is concerned with the role of data in the design, development, management and utilization of information systems.

- Data Management Systems and Modern Hardware/Software Platforms
- Data Models, Data Integration, Semantics and Data Quality
- Spatial, Temporal, Graph, Scientific, Statistical and Multimedia Databases
- Data Mining, Data Warehousing, and OLAP
- Big Data, Streams and Clouds
- Information Management, Distribution, Mobility, and the WWW
- Data Security, Privacy and Trust
- Performance, Experiments, and Analysis of Data Systems

The TCDE sponsors the International Conference on Data Engineering (ICDE). It publishes a quarterly newsletter, the Data Engineering Bulletin. If you are a member of the IEEE Computer Society, you may join the TCDE and receive copies of the Data Engineering Bulletin without cost. There are approximately 1000 members of the TCDE.

Join TCDE via Online or Fax

ONLINE: Follow the instructions on this page:

www.computer.org/portal/web/tandc/joinatc

FAX: Complete your details and fax this form to **+61-7-3365 3248**

Name ____________________

IEEE Member # ____________________

Mailing Address ____________________

____

Country ____________________

Email ____________________

Phone ____________________

TCDE Mailing List

TCDE will occasionally email announcements, and other opportunities available for members. This mailing list will be used only for this purpose.

Membership Questions?

Xiaoyong Du
Key Laboratory of Data Engineering
and Knowledge Engineering
Renmin University of China
Beijing 100872, China
duyong@ruc.edu.cn

TCDE Chair

Xiaofang Zhou
School of Information Technology and
Electrical Engineering
The University of Queensland
Brisbane, QLD 4072, Australia
zxf@uq.edu.au

IEEE Computer Society
10662 Los Vaqueros Circle
Los Alamitos, CA 90720-1314