Hyper Questions: Unsupervised Targeting of a Few Experts in Crowdsourcing

Jiyi Li
Kyoto University, Japan
jyli@i.kyoto-u.ac.jp

Yukino Baba
Kyoto University, Japan
baba@i.kyoto-u.ac.jp

Hisashi Kashima
Kyoto University, Japan
RIKEN Center for AIP, Japan
kashima@i.kyoto-u.ac.jp

ABSTRACT

Quality control is one of the major problems in crowdsourcing. One of the primary approaches to rectify this issue is to assign the same task to different workers and then aggregate their answers to obtain a reliable answer. In addition to simple aggregation approaches such as majority voting, various sophisticated probabilistic models have been proposed. However, given that most of the existing methods operate by strengthening the opinions of the majority, these models often fail when the tasks require highly specialized knowledge and the ability of a large majority of the workers is inadequate. In this paper, we focus on an important class of answer aggregation problems in which majority voting fails and propose the concept of *hyper questions* to devise effective aggregation methods. A hyper question is a set of single questions, and our key idea is that experts are more likely to provide correct answers to all of the single questions included in a hyper question than non-experts. Thus, experts are more likely to reach consensus on the hyper questions than non-experts, which strengthen their influences. We incorporate the concept of hyper questions into existing answer aggregation methods. The results of our experiments conducted using both synthetic datasets and real datasets demonstrate that our simple and easily usable approach works effectively in cases where only a few experts are available.

CCS CONCEPTS

• **Information systems → Data mining**;

KEYWORDS

Crowdsourcing; Answer aggregation; Heterogeneous-Answer Multiple-Choice questions; Hyper Question

1 INTRODUCTION

Crowdsourcing offers easy and on-demand access to human intelligence at scale and has been successfully applied to various scientific and business areas. One typical crowdsourcing task is multiple-choice questions, in which workers are asked to select one answer from multiple candidates for a given question, such as a tag of an image or an answer to a scientific question. However, quality control is a major problem in crowdsourcing because crowdsourcing workers often fail to provide correct answers owing to various reasons, such as lack of ability, mistakes, and laziness.

Q. Which of the following drugs is most likely to cause Cushing's syndrome with long-term use?

(a) Heparin, (b) Insulin, (c) Theophylline, (d) Prednisolone

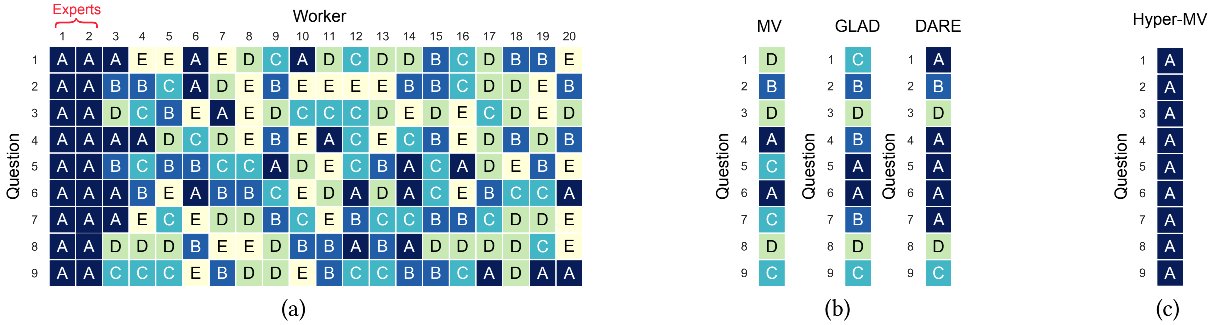
Figure 1: Example of a difficult medical question that was used in our experiments

One of the major approaches to rectify this issue is to introduce redundancy, i.e., assigning the same task to different workers and aggregating their answers to obtain a reliable answer. In addition to simple aggregation approaches such as majority voting, various sophisticated statistical methods have been proposed [1, 2, 6, 10, 16, 19–22, 24]. Most of these methods are based on the assumption that high-ability workers are likely to give correct answers. However, given that worker ability and true answers are both unknown, they are usually estimated in a reciprocal manner. In most approaches, workers who provide majority answers are regarded as high-ability workers; specifically, the opinions of the majority are strengthened. Consequently, most approaches often fail when the majority of workers provide wrong answers. These cases occur when tasks require highly specialized knowledge and the ability of a large majority of the workers is inadequate. For example, difficult medical questions can be correctly answered by only a few medical experts existing in crowds. An example of these questions is given in Figure 1. Practical motivations for asking such medical questions in crowdsourcing platforms are to construct a large scale medical knowledge base or to annotate a medical dataset for training machine learning models. Oosterman et al. [13] made an exploration study which shows that crowdsourcing has the potential for providing useful annotations for the knowledge intensive tasks, while the difficulty of tasks clearly played a role in the label performance.

Let us consider a motivating example. Figure 2(a) shows the answers given by 20 workers to nine five-choice questions. Among the 20 workers, only two (Workers 1 and 2) are experts who always give the correct answers, and the other 18 are non-experts who give random answers. The experts are overwhelmed by the large number of non-experts; therefore, both majority voting and statistical method (GLAD [22]) completely fail. Furthermore, DARE [1],

CIKM'17, November 6–10, 2017, Singapore.

ACM ISBN 978-1-4503-4918-5/17/11…$15.00
https://doi.org/10.1145/3132847.3132971


Experts
Worker
Question
MV
GLAD
DARE
Hyper-MV

Question	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	A	A	A	E	E	A	E	D	C	A	D	C	D	D	B	C	D	B	B	E
2	A	A	B	B	C	A	D	E	B	E	E	E	E	B	B	C	D	D	E	B
3	A	A	D	C	B	E	A	E	D	C	C	C	D	E	D	E	C	D	E	D
4	A	A	A	A	D	C	D	E	B	E	A	C	E	C	B	E	D	B	D	B
5	A	A	B	C	B	B	C	C	A	D	E	C	B	A	C	A	D	E	B	E
6	A	A	A	B	E	A	B	B	C	E	D	A	D	A	C	E	B	C	C	A
7	A	A	A	E	C	E	D	D	B	C	E	B	C	C	B	B	C	D	D	E
8	A	A	D	D	D	B	E	E	D	B	B	A	B	A	D	D	D	D	C	E
9	A	A	C	C	C	E	B	D	D	E	B	C	C	B	B	C	A	D	A	A

(a)

Question	MV	GLAD	DARE
1	D	C	A
2	B	B	B
3	D	D	D
4	A	B	A
5	C	A	A
6	A	A	A
7	C	B	A
8	D	D	D
9	C	C	C

(b)

Question	Hyper-MV
1	A
2	A
3	A
4	A
5	A
6	A
7	A
8	A
9	A

(c)

Figure 2: Motivating example: (a) Twenty workers answered nine five-choice questions; most of the workers are non-experts who give random answers while only two experts give correct answers ('A' is the correct answer for all of the questions). (b) Both majority voting and other statistical approaches (GLAD [22] and DARE [1]) fail, whereas (c) the proposed method (Hyper-MV) successfully obtains the correct answers.

which is one of the most sophisticated statistical methods based on Bayesian inference, does not perform satisfactorily (Figure 2(b)).

Our key solution to the problem is to focus on sets of questions (which we call *hyper questions*) instead of individual questions. We observe that experts are more likely to give correct answers to all question of a particular set than the non-experts. Although there is not a small chance for non-experts to randomly guess the correct label of an individual question, it is much more difficult for them to guess the correct labels to a set of two or more questions; in contrast, it is not difficult for an expert on the same set of single questions because he knows the answer. Specifically, the difference between the accuracies of experts and those of non-experts for a set of questions is greater than that for a single question, which makes experts easier to reach consensus than the non-experts and thus more likely to become the majority. Based on this underlying concept, we propose a simple and effective solution called Hyper-MV that takes into account majority voting on hyper questions. The proposed solution enables experts to reach agreements more easily than non-experts and strengthens their influences. Figure 2(c) shows that our approach can generate the correct aggregation results for all questions in this example. The notion of hyper questions can also be applied to methods other than majority voting; for example, it can be combined with the GLAD method or the DARE method. We incorporate hyper questions into GLAD and DARE, and propose Hyper-GLAD and Hyper-DARE.

The results of experiments conducted using both synthetic and real datasets indicate that our approach achieves high accuracies for cases where only a few experts are available. We found that our approach is especially ideal for the scenario in which capable experts are first found by asking all workers to answer a small set of questions, and then asking the found experts to work on the remaining questions. This scenario has been already employed in the previous work [9].

The contributions of this paper are threefold:

(1) We discuss an important class of answer aggregation problems for multiple choice questions with heterogeneous answers in which majority voting fails.
(2) We propose the notion of hyper questions that can efficiently increase the accuracy difference between non-experts and experts.
(3) We incorporate the concept of hyper questions into the existing aggregation methods, and propose Hyper-MV, Hyper-GLAD, and Hyper-DARE, which can strengthen the influences of experts to obtain better aggregation results. Our simple and easily usable approach works effectively in cases where only a few experts are available.

2 RELATED WORK

Majority voting is a widely used aggregation method that assigns equal weights to all workers. Given that workers have diverse abilities, researchers seek to find high-ability workers to improve aggregation performance. Some approaches jointly estimate worker ability and true answers using the expectation-maximization (EM) algorithm [2, 16, 22], bipartite models [6], the maximum entropy principle [24], or Bayesian inference [1, 8, 10, 19–21]. Zheng et al. reviewed and compared the existing work on label aggregation in the scenario of truth discovery [23]. Generally, most of these approaches tend to strengthen the opinions of the majority workers, and therefore do not perform well in cases where the number of capable workers (i.e., experts) is smaller than the number of incapable workers.

Only a few studies have addressed the few-expert scenario. Li et al. used worker profiles, such as demographic information to predict their abilities [9]. Ma et al. focused on task information, such as the words appearing in task descriptions, and combined a topic model and a worker ability model to characterize worker expertise [11]. Kazai et al. used demographics information and personality traits [7]. Ipeirotis et al. implemented a version of D&S [2] by using golden examples [5]. Ipeirotis and Gabrilovich used sponsored search associated with medical terms to guide medical experts to medical question tasks [4]. Prelec et al. proposed the use of the response for an additional question asking the distribution of other people's answers [15]. This approach considers an answer is more likely to be correct when the answer is more popular than people predicted. Mortensen et al. [12] utilized crowdsourcing for constructing biomedical ontologies and selected proper workers with biology knowledge by qualification test. In contrast to these methods, our approach uses only the worker answers without auxiliary information, because it is often unavailable. The application of qualification tests is another approach for targeting the expert workers [3]; however, it is not always guaranteed that well-designed qualification tests are available for requesters.

Although the worker ability may somewhat increase when answering more questions [14], because our questions are knowledge-intensive difficult questions, we regard that non-experts cannot become experts by only answering a limited number of questions. Modeling the change of worker ability is not in the scope of this paper.

Our approach assumes correlations among answers by experts for a set of questions. This is similar to the assumption that class labels are correlated in multi-label classification problems. Our method is conceptually similar to RA*k*EL [18], which considers a set of randomly selected class variables as one super-class variable.

3 PROBLEM DEFINITION

We address a typical crowdsourcing task consisting of multiple-choice questions. A multiple-choice question has several candidate choices for the answer and the crowd workers are asked to select one of the given choices. There are at least two types of multiple-choice questions: homogeneous-answer questions and heterogeneous-answer questions.

Homogeneous-answer questions have the same candidate answers for all questions. An example of such questions is the yes-no question. For example, in an image annotation task, given a set of images with cat or not, all questions for these images can be "*Can you see a cat in the image?*", where one candidate answer is always 'yes' and the other is always 'no.'

In contrast, *heterogeneous-answer questions* can have the different candidate answers for each question. For example, in a task which asks the words with same meaning in a different language, one question can be "*How to say 'yes' in German? (a) Ja, (b) Nein*" and another question can be "*How to say 'thank you' in German? (a) Bitte, (b) Danke.*"

In this study, we focus on the heterogeneous-answer multiple-choice questions.

Given that the ability and motivation of workers are not constant, the answers of some workers can be more reliable than those of the other workers. Therefore, we usually ask the same question to different workers and aggregate their answers to obtain a reliable answer. We assume that there is a set of workers $\mathcal{W}$ and a set of questions Q. Each question $q \in Q$ has a set of candidate answers C_q, from which we ask each worker to select one answer. We denote the set of all answers by $\mathcal{L} = \{l_{wq}\}_{w \in \mathcal{W}, q \in Q}$.

In this paper, we especially focus on *difficult questions* that require specialized knowledge in which only a few experts are capable of giving the correct answers while most non-experts are incapable. The serious difficulty in this situation is that simple majority voting fails to agree with the correct answers. Many statistical methods that are more sophisticated than simple majority voting have been proposed. Most of these methods are based on the assumption that high-ability workers are likely to give correct answers. Because the worker ability and true answers are both unknown, they are usually estimated in a reciprocal manner. However, in most of these methods, workers who provide majority answers are regarded as high-ability workers; specifically, the opinions of the majority are strengthened. Consequently, these methods often fail when the majority of the workers provide wrong answers.

Definition 3.1. Problem Definition: given the worker set $\mathcal{W}$ with only a few capable experts, the heterogeneous-answer question set Q, and the answer set $\mathcal{L}$, we aim to estimate the true answer for each question $q \in Q$.

4 HYPER QUESTIONS AND ANSWER AGGREGATION METHODS

4.1 Hyper Questions

When the ability of a single worker and the correctness of his/her answers are not secured, we redundantly assign the same question to multiple workers, and aggregate their answers to obtain a more reliable answer. Majority voting is a common aggregation approach. However, this approach fails when the question is too difficult and the correct answer fails to obtain the majority of the votes. To find the few capable workers that are outnumbered by the non-capable workers, we propose the concept of *hyper questions.*

Definition 4.1. k-hyper question: A hyper question consists of a subset of original single questions, and an answer to a hyper question is a set of answers to the questions included in the hyper question. A set of k original single questions is defined as a *k-hyper question.*

The belief underlying hyper questions is that experts are more likely to give correct answers to all questions in particular set of questions than non-experts. Specifically, the difference between the accuracies of experts and those of non-experts for a set of questions is greater than that for a single question, which makes the experts more likely to win in majority voting.

Let us examine how a hyper question works by using an extreme example with perfect experts and random non-experts. For two-choice questions, the experts always give correct answers with 100% accuracy, and the non-experts give random answers that result in 50% accuracy. The correct answers by two experts always coincide, while the answers given by two non-experts have a 25% probability of coinciding with the incorrect label by chance for a single two-choice question. When the number of non-experts is large, the expected number of these undesirable coincidences will increase; therefore, the wrong answers will have more chance of winning the majority. On the other hand, if we consider a 2-hyper question having a pair of two-choice single questions, the number of choices is four; therefore, the probability of a particular undesirable coincidence will decrease to 6.3%. With a 3-hyper question, the probability will be further decreased to 1.6%. With hyper questions, we can increase the difference in performance between two worker groups with different levels of ability and differentiate the experts from the non-experts more easily.

4.2 Answer Aggregation on Hyper Questions

We then incorporate the concept of hyper questions into answer aggregation methods. We start by presenting Hyper-MV, which takes majority voting on hyper questions.

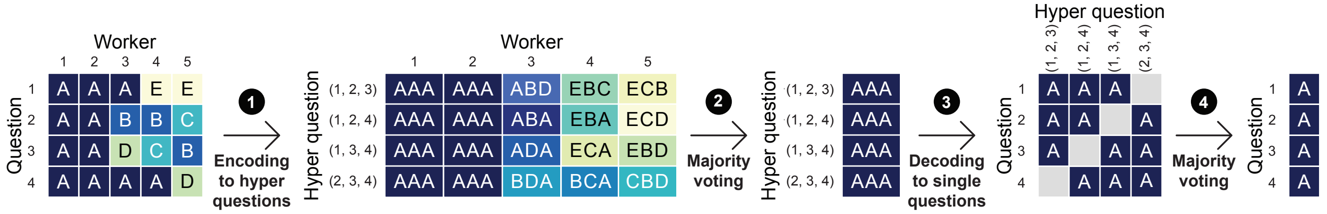


Figure 3: Example of Hyper-MV procedure: (1) The $k(=3)$-hyper questions are generated from the original single questions. (2) Majority voting is applied to each hyper question. (3) The results of the majority voting are decoded to votes on individual questions. (4) Another round of majority voting aggregates the votes to obtain the final answers.

```
Algorithm 1: Hyper-MV
  Input: Worker set $\mathcal{W}$; Question set $\mathcal{Q}$; Answer set $\mathcal{L}$;
  Output: Estimated true answers $\{z_q\}_q$;
  Parameters: Size of hyper question $k$;
  Generate a set of hyper questions: $\mathcal{S} = \{(q_{u1}, \ldots, q_{uk})\}_u$;
  foreach $s_u \in \mathcal{S}$ do
      foreach $w \in \mathcal{W}$ do
          Obtain the worker's answer to the $u$-th hyper
            question: $\boldsymbol{x}_{wu} = (l_{wq_{u1}}, \ldots, l_{wq_{uk}})$;
      end
  end
  foreach $s_u \in \mathcal{S}$ do
      $\boldsymbol{x}_u$ = MajorityVoting $(\{\boldsymbol{x}_{wu}\}_w)$;
  end
  foreach $s_u \in \mathcal{S}$ do
      for $v = 1, \ldots, k$ do
          Obtain an answer to the single question $q_{uv}$:
            $y_{uq_{uv}} = x_{uv}$;
      end
  end
  foreach $q \in \mathcal{Q}$ do
      $z_q$ = MajorityVoting $\left(\{y_{uq}\}_u\right)$;
  end
  return $\{z_q\}_q$
```

Hyper-MV first constructs a set of hyper questions by combining single questions in $\mathcal{Q}$. Constructing all possible k-hyper questions from $\mathcal{Q}$ results in $\binom{|\mathcal{Q}|}{k}$ hyper questions. A k-hyper question consisting of questions $q_1, q_2, \ldots, q_k$ has $\prod_{\kappa=1}^{k} |C_{q_\kappa}|$ choices. Each answer to a hyper question is represented by the concatenation of the answers in the single questions included in the hyper question.

Hyper-MV then performs the majority voting to each hyper question, which results in an answer to the hyper question. The aggregated results of the hyper questions are decoded into answers of the single questions. Finally, another round of majority voting is carried out for each single question. Consequently, the results of the first round of majority voting on hyper questions are aggregated to obtain the final answer for each single question.

Figure 3 shows the procedure of Hyper-MV, which consists of five workers and four single questions, and k is set to 3. 'A' is the correct answer for all of the single questions. In the first step, four 3-hyper questions are created from the four single questions. An answer to a hyper question is the concatenation of the answers to the constituent single questions. In the second step, Hyper-MV applies majority voting to each hyper question; in this case, the answer 'AAA' is chosen for all the hyper questions. In the third step, each of the majority answers for the hyper questions votes for the single questions included in it. Finally, in the fourth step, another round of majority voting aggregates the votes to the single questions to obtain the final answers.

Algorithm 1 formulates the Hyper-MV approach. Hyper-MV first generates a set of k-hyper questions, $\mathcal{S}$, where the u-th hyper question $\boldsymbol{s}_u$ consists questions $q_{u1}, \ldots, q_{uk}$. The answer to the u-th hyper question given by the worker w is represented by $\boldsymbol{x}_{wu} = (l_{wq_{u1}}, \ldots, l_{wq_{uk}})$, where $l_{wq_{uv}}$ is the answer to the question q_{uv} given by the worker w. Hyper-MV next performs the majority voting to each hyper question s_u to obtain $\boldsymbol{x}_u = (x_{u1}, \ldots, x_{uk})$. Note that MajorityVoting $(\mathcal{A})$ in the Algorithm 1 returns the most frequent element in the given set $\mathcal{A}$. Each aggregated result, $\boldsymbol{x}_u$, is then decoded to $y_{uq_{u1}}, \ldots, y_{uq_{uk}}$, where $y_{uq_{uv}}$ is an answer to the single question q_{uv}, and then another round of majority voting aggregates the answers to obtain the final output $\{z_q\}_q$.

The concept of hyper questions is itself independent of underlying aggregation methods; therefore, majority voting is not the only method that can benefit from hyper questions. The first round of majority voting on hyper questions can be replaced with other aggregation methods, such as GLAD [22] or DARE [1]. GLAD models the worker ability and question difficulty in the probability that a label is correct and utilizes EM algorithm to estimate the true answers. We denote our proposed method based on GLAD as Hyper-GLAD. DARE has a Bayesian graphical model on the worker ability and question difficulty. We denote our proposed method based on DARE as Hyper-DARE. Although there are various existing methods for aggregating crowdsourcing answers, we selected GLAD and DARE because they were designed for the heterogeneous-answer questions. The existing work designed for the homogeneous-answer questions such as D&S [2], BCC[8] and CommunityBCC [19] construct the confusion matrix of workers on the same candidate categorical answers in all questions, and thus are not proper for the heterogeneous-answer questions.

Our approach is proposed to generate the aggregated answers. It actually can be used to select the candidate experts. In the unsupervised scenario, the aggregated answers can be used as pseudo true answers to estimate the worker accuracy. The workers with high accuracies can be regarded as experts and selected for much more questions.

4.3 Random Sampling

When the number of hyper question is quite huge. it is time consuming and sometimes almost impossible to consider all possible hyper questions even for several dozen of single questions. For example, the number of 6-hyper questions from 60 single questions exceeds 50 million. We resort to random sampling to accelerate the answer aggregation on hyper questions: shuffle the order of all single questions uniformly at random to generate a permutation; in this permutation, pick every k single questions from the beginning of the queue to generate k-hyper questions as long as we can pick them; carry out the shuffling and picking for r times. This procedure generates $r \cdot \lfloor \frac{|Q|}{k} \rfloor$ hyper questions in total. This sampling method has an advantage that all single questions have almost similar occurrences in the set of hyper questions. In Section 5.2.3, we will discuss the performance influences by the parameter selection on the sampling size.

5 EXPERIMENTS

5.1 Experiments on Synthetic Datasets

5.1.1 Datasets. We conducted experiments using synthetic datasets to validate whether Hyper-MV is superior to majority voting, especially in cases where the capability of most of the workers is insufficient for the given tasks. Different numbers of experts, as well as different expert ability values, were tested to determine the conditions under which Hyper-MV performs well. We set the number of workers to twenty and assumed there were twenty five-choice questions. We generated 100 datasets with different (n_e, p_e) values, with the number of experts n_e varying from $\{2, 4, 6\}$ and the probability p_e of an expert giving a correct answer varying from $\{0.8, 0.9, 0.95, 1.0\}$. The answers by the non-expert workers were randomly selected from the five choices.

5.1.2 Methods. We compared Hyper-MV ($k = 5$) with majority voting on single questions. In the random sampling settings, we generated 100 random permutations of the single questions and generated four 5-hyper questions from each of the permutations.

5.1.3 Results. Table 1 shows the average accuracies (i.e., the ratio of correct answers) of each method for 100 datasets. Hyper-MV robustly outperformed the majority voting, especially when the number of experts is small. When more than 20% of the workers are experts, the Hyper-MV outputted accurate answers even when the accuracies of the experts are not very high.

5.2 Experiments on Real Datasets

The real datasets used in the existing work (e.g., [17, 23]) contain relatively high ability workers and homogeneous-answer questions. To verify our proposal in our scenario, we thus create some real datasets which have relatively difficult heterogeneous-answer questions and a few experts with various settings. They have various numbers of choices and numbers of questions. The scale of number of choices is consistent with the datasets in existing work. The scale of number of questions follows the scenario settings in this work in which capable experts are first found by asking all workers to answer a small set of questions.

5.2.1 Datasets. We also investigated the efficacy of Hyper-MV, Hyper-GLAD, and Hyper-DARE using real datasets. We constructed real crowdsourcing datasets using Lancers,[1] a commercial crowdsourcing platform in Japan. During the collection of the datasets, all workers were asked to answer all questions, and both the questions and the candidate answers were shown to each worker in random order (based on the Discussion section).

We prepared six real datasets comprising the answers for questions on the following topics: *Chinese language*, *English language*, *Information technology (IT)*, *medicine*, *Pokémon*, and *science*. The questions in the *Chinese* dataset concern the meaning of Chinese vocabularies. The *English* questions were collected from the verbal section in the Graduate Record Examinations (GRE), which ask subjects to select an analogical word pair having the most similar relationship to a given word pair. The *IT* questions are concerned with the basic knowledge of information technology, which were obtained from the Japan Information-Technology Engineers Examination. The *Medicine* questions were taken from a national examination for nurses. Questions on drug efficacy and side effects were included in the dataset. The *Pokémon* dataset contains the questions asking the Japanese name of a given Pokémon with the English name. The *Science* questions concern intermediate knowledge of chemistry and physics.

All questions in the datasets were heterogeneous-answer questions; that is, each question had different candidate answers. The answers in each dataset were provided by a different set of workers and were summarized in Table 2. [2] Figure 4 shows the distribution of the worker accuracies in each real dataset. Given that the questions required some sort of specialized knowledge, a relatively small number of workers have high accuracies in most of the datasets.

5.2.2 Methods. We compared Hyper-MV, Hyper-GLAD, and Hyper-DARE with majority voting on single questions (MV), GLAD, and DARE, respectively. We randomly sampled the hyper questions when we performed Hyper-MV, Hyper-GLAD, and Hyper-DARE using the same sampling procedure described in Section 5.1.1. We applied our hyper-question based methods to 10 different sets of hyper questions. We set the priors of GLAD to $\alpha \sim \mathcal{N}(1, 1)$ and $\beta' \sim \mathcal{N}(1, 1)$, and those of DARE to $a_p \sim \mathcal{N}(0, 1)$, $d_q \sim \mathcal{N}(0, 1)$, and $\tau_q \sim \text{Gamma}(1, 0.01)$. We varied the size of hyper questions (i.e., k) from two to seven for Hyper-MV, and from two to four for Hyper-GLAD and Hyper-DARE. Because we focus on proposing the answer aggregation method in the unsupervised scenario which is the common problem settings in the answer aggregation methods like GLAD and DARE, we did not compare with the existing methods which focus on proposing additional strategy of qualification test and using auxiliary information.

5.2.3 Results. Table 3 shows the comparisons between the majority voting and Hyper-MV, GLAD and Hyper-GLAD, and DARE and Hyper-DARE. In all the datasets, Hyper-MV with $k = 4, 5, 6$

[1] http://www.lancers.jp

[2] These datasets are available at http://www.ml.ist.i.kyoto-u.ac.jp/en/en-research/li2017cikm

Table 1: Averages and standard deviations of accuracies on synthetic datasets according to the number of experts (n_e, out of 20) and the probability of an expert giving a correct answer (p_e). Statistically significant ($p < 0.05$) winners by the Wilcoxon signed rank test are underlined. Hyper-MV achieves better performance than the majority voting (MV) especially when the number of experts is small.

p_e	$n_e = 2$		$n_e = 4$		$n_e = 6$	
	MV	Hyper-MV ($k = 5$)	MV	Hyper-MV ($k = 5$)	MV	Hyper-MV ($k = 5$)
0.80	0.398 ±0.117	<u>0.597</u> ±0.182	0.629 ±0.113	<u>0.929</u> ±0.072	0.840 ±0.077	<u>0.974</u> ±0.038
0.90	0.446 ±0.098	<u>0.838</u> ±0.133	0.722 ±0.092	<u>0.989</u> ±0.028	0.916 ±0.068	<u>0.998</u> ±0.010
0.95	0.460 ±0.112	<u>0.932</u> ±0.058	0.759 ±0.086	<u>0.999</u> ±0.007	0.945 ±0.055	<u>1.000</u> ±0.005
1.00	0.475 ±0.116	<u>1.000</u> ±0.000	0.809 ±0.090	<u>1.000</u> ±0.000	0.977 ±0.035	<u>1.000</u> ±0.000

Table 2: Summary of real datasets

Dataset	#questions	#workers	#choices
Chinese	24	50	5
English	30	63	5
IT	25	36	4
Medicine	36	45	4
Pokémon	20	55	6
Science	20	111	5

or 7 outperformed the majority voting. We found that the performance of Hyper-MV is rather inferior when k is small, which indicates that the hyper questions constructed from a small number of single questions are not sufficient to strengthen the influences of the experts. We also conducted experiments using Hyper-MV with $k \geq 8$ and observed that its performance declined when k was large. One of the reasons for this scenario is that the experts in the real datasets do not always provide correct answers, and thus their answers for a large size of hyper question can easily be mismatched.

Hyper-GLAD and Hyper-DARE achieved better performance than GLAD and DARE, respectively, in almost all the cases. This result indicates that incorporating hyper questions is beneficial for improving the worker ability estimation of GLAD and DARE, even if k is small.

Hyper-MV with $k = 5, 6$, or 7 demonstrated better or competitive performance with GLAD and DARE in many cases; it is notable that the performance of this simple method is on par with the state-of-the-art method with a sophisticated graphical model and Bayesian inference. Based on the above observations, for cases where all the workers answer on all the questions, we suggest to use a moderate size of hyper questions for Hyper-MV. A small size of hyper question is sufficient for Hyper-GLAD and Hyper-DARE.

To investigate the robustness of the methods to missing answers, we randomly sampled $r\%$ of the answers in the real datasets and then applied the methods. For each r, we performed the random sampling ten times and applied the methods to each sampled answers. When we construct answers for hyper questions, an answer for a hyper question is considered as not given if the answer for at least one single question included in the hyper question is missing. The average accuracies among the ten sampling results are shown in Figure 6. When the missing ratio is less than or equal to 0.3, Hyper-MV ($k = 3, 5$), Hyper-GLAD ($k = 3$), and Hyper-DARE ($k = 3$) perform better than MV, GLAD, and DARE, respectively, in most cases. However, their performances decline with increasing ratio of missing answers. This is because our methods need the experts to answer on the same hyper questions to distinguish them from the non-experts, and it becomes less likely when there are many missing answers and the size of each hyper

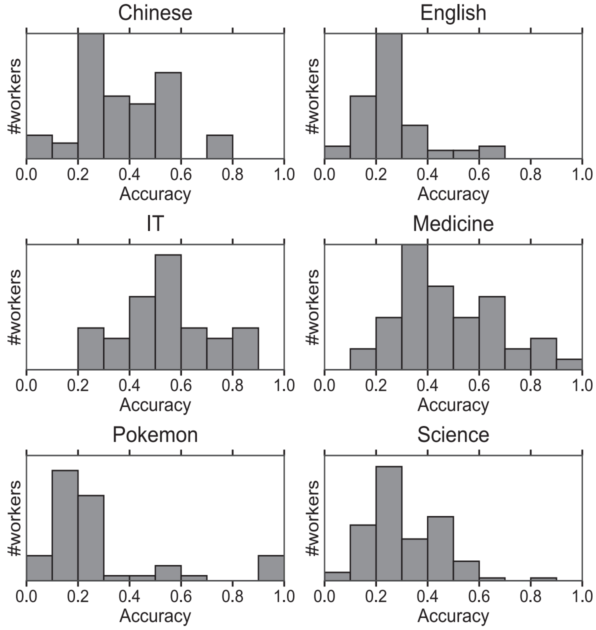


Figure 4: Distributions of worker accuracies in the real datasets; a relatively small number of experts are available in the datasets.

Table 3: Comparisons between the majority voting (MV) and Hyper-MV, GLAD and Hyper-GLAD, and DARE and Hyper-DARE. Averages and standard deviations of the accuracies of Hyper-MV, Hyper-GLAD, and Hyper-DARE over 10 different sets of hyper questions are shown. The cases where the proposed method outperformed the competitor are underlined. Our hyper question-based methods achieved better performance than the corresponding existing methods in almost all the cases.

(a) MV vs. Hyper-MV

Dataset	MV	Hyper-MV					
		$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$
Chinese	0.625	0.604 ±0.021	0.617 ±0.017	0.642 ±0.038	0.637 ±0.019	0.671 ±0.035	0.646 ±0.034
English	0.433	0.460 ±0.020	0.513 ±0.031	0.523 ±0.042	0.597 ±0.046	0.563 ±0.046	0.497 ±0.057
IT	0.720	0.772 ±0.031	0.804 ±0.022	0.812 ±0.018	0.788 ±0.018	0.796 ±0.022	0.800 ±0.018
Medicine	0.667	0.747 ±0.019	0.803 ±0.015	0.850 ±0.022	0.864 ±0.026	0.900 ±0.022	0.894 ±0.017
Pokémon	0.650	0.975 ±0.034	1.000 ±0.000	1.000 ±0.000	1.000 ±0.000	1.000 ±0.000	1.000 ±0.000
Science	0.550	0.555 ±0.015	0.580 ±0.033	0.620 ±0.024	0.620 ±0.033	0.605 ±0.035	0.650 ±0.071

(b) GLAD vs. Hyper-GLAD

Dataset	GLAD	Hyper-GLAD		
		$k = 2$	$k = 3$	$k = 4$
Chinese	0.542	0.696 ±0.053	0.775 ±0.028	0.750 ±0.032
English	0.567	0.663 ±0.023	0.713 ±0.022	0.730 ±0.028
IT	0.720	0.840 ±0.000	0.820 ±0.020	0.824 ±0.020
Medicine	0.694	0.889 ±0.000	0.917 ±0.012	0.931 ±0.014
Pokémon	0.850	1.000 ±0.000	1.000 ±0.000	1.000 ±0.000
Science	0.550	0.645 ±0.061	0.645 ±0.042	0.690 ±0.037

(c) DARE vs. Hyper-DARE

Dataset	DARE	Hyper-DARE		
		$k = 2$	$k = 3$	$k = 4$
Chinese	0.625	0.658 ±0.025	0.667 ±0.046	0.708 ±0.019
English	0.600	0.677 ±0.026	0.680 ±0.031	0.690 ±0.021
IT	0.800	0.816 ±0.020	0.804 ±0.012	0.828 ±0.018
Medicine	0.861	0.833 ±0.000	0.939 ±0.021	0.956 ±0.014
Pokémon	1.000	1.000 ±0.000	1.000 ±0.000	1.000 ±0.000
Science	0.600	0.600 ±0.000	0.605 ±0.015	0.640 ±0.054

question (i.e., k) is large. This is supported by the comparison between Hyper-MV with $k = 3$ and that with $k = 5$. The decline in the accuracies of Hyper-MV with $k = 3$ according to the missing answer ratio is more gradual than that of Hyper-MV with $k = 5$.

We conducted additional experiments to examine the accuracies of our methods with a small number of questions. For each of our real datasets, we randomly selected ten single questions and applied Hyper-MV ($k = 5$), Hyper-GLAD ($k = 3$), and Hyper-DARE ($k = 3$) to the answers on the questions to compare the accuracies with the corresponding existing methods. The averages and standard deviations of accuracies among 100 different sets of single questions are shown in Table 4. Even when the number of questions is small, Hyper-MV, Hyper-GLAD, and Hyper-DARE outperformed their corresponding existing methods in most of the datasets. Thus, we conclude that our hyper question approaches are suitable for scenarios in which a small number of questions are used to discover experts by asking all workers to answer all questions.

The number of sampled hyper questions may affect the accuracies of our methods. We conducted experiments with varying the number of hyper questions. Specifically, we investigated the accuracies of Hyper-MV ($k = 5$) with different numbers of the hyper questions. The results are as shown in Figure 5. In all the datasets except *English* and *Science*, the accuracies did not decrease even with the small number of hyper questions. Given that the numbers of the experts in the *English* and *Science* datasets are smaller than those in other datasets, we may require a moderate number of hyper questions to strengthen the influences of the experts. However, the accuracies on the two datasets were stable when the number of hyper questions are more than or equal to 200.

Table 4: Accuracies on real datasets with the small number of questions. The number of questions is set to 10. Averages and standard deviations of the accuracies among the 100 different sets of questions are shown. Statistically significant ($p < 0.05$) winners by the Wilcoxon signed rank test are underlined. Hyper-MV, Hyper-GLAD, and Hyper-DARE outperform the majority voting (MV), GLAD, and DARE in most of the datasets.

Dataset	MV	Hyper-MV (k=5)	GLAD	Hyper-GLAD (k=3)	DARE	Hyper-DARE (k=3)
Chinese	0.614 ±0.125	0.633 ±0.152	0.591 ±0.142	0.635 ±0.204	0.620 ±0.172	0.635 ±0.189
English	0.442 ±0.132	0.551 ±0.192	0.476 ±0.160	0.582 ±0.217	0.521 ±0.208	0.560 ±0.222
IT	0.723 ±0.122	0.791 ±0.110	0.741 ±0.120	0.800 ±0.112	0.789 ±0.115	0.795 ±0.114
Medicine	0.640 ±0.128	0.864 ±0.128	0.679 ±0.134	0.887 ±0.139	0.768 ±0.161	0.853 ±0.153
Pokémon	0.651 ±0.104	1.000 ±0.000	0.849 ±0.084	1.000 ±0.000	0.980 ±0.113	1.000 ±0.000
Science	0.569 ±0.118	0.641 ±0.119	0.579 ±0.120	0.350 ±0.288	0.613 ±0.121	0.631 ±0.148

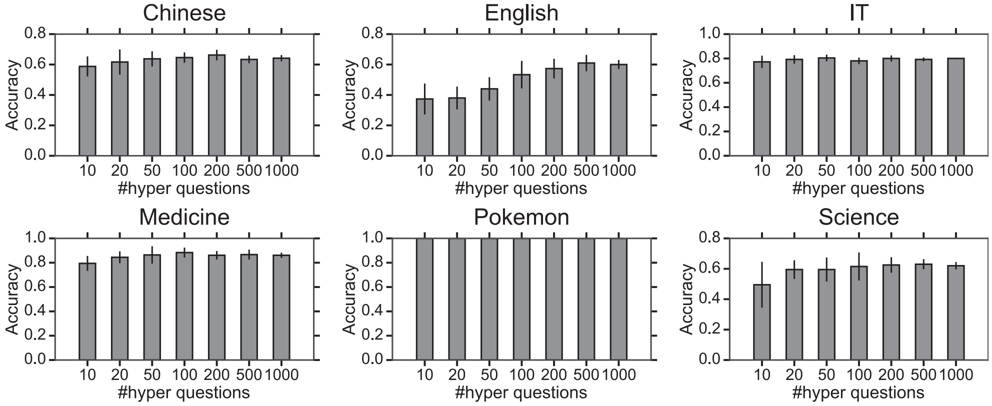


Figure 5: Accuracies of Hyper-MV with $k = 5$ on real datasets according to the number of sampled hyper questions. Averages and standard deviations of the accuracies among 10 different sets of hyper questions are shown. The accuracies are stable in all the datasets when the number of hyper questions is more than or equal to 200.

6 DISCUSSION

6.1 Missing Answers

Hyper question is definitely not a versatile solution. We discuss its several limitations and solutions to some of them below. As shown in the experiments (Figure 6), our approach expects workers to give answers to most questions, which can be a limitation in some cases. On the other hand, the results shown in Table 4 support that our approach is especially applicable to scenarios in which capable experts are first found by asking all workers to answer completely a small set of questions, and then asking the found experts to work on the remaining questions [9]. This is a feasible approach for targeting experts in practical situations.

6.2 Biases

Our approaches are based on the assumption that expert answers coincide across different questions while those by non-experts do not. However, it is possible that wrong answers also coincide by chance or as a result of biases, e.g., the collusion of malicious workers. We categorize answer biases into *contextual biases* and *semantic biases*. Contextual biases occur depending on the context in which choices are presented and regardless of the choices themselves. Typical examples are the biases introduced by the order of presentation of questions and the layout of choices. On the other hand, semantic biases occur according to the actual contents of the presented choices. A possible simple solution to cope with the biases is to randomize the order of the questions and choices presented to each worker. This randomization can reduce the contextual biases, but it is not effective for semantic biases.

6.3 Incapable, Spam, and Malicious Workers

To investigate the biases possibly induced in our setting, we categorize potentially harmful workers into four categories:

- *Incapable (but diligent) workers*, who try their best even with difficult questions. Some of their efforts end up in random answers, which results in contextual biases. However, when a question is likely to induce some particular misunderstanding, they tend to give a particular wrong answer to the question, which results in semantic biases. Hyper questions decrease the effect of semantic biases unless all of the single questions included in a hyper question are of this type.

- *Spam workers*, who always choose answers without regard to the given choices. Workers in this category always give random answers or choose answers in the same position (e.g., the first choice). By definition, they can introduce only contextual biases.
- *Malicious non-experts*, who do not know the correct answers, but try to have wrong answers obtain the majority of votes. Unless they collude, they can only behave similar to spam workers and introduce only contextual biases.
- *Malicious experts*, who know the correct answers but try to cause disruption. They have more control on semantic biases. If the number of choices is only two, the wrong answers by the malicious experts will coincide. To avoid this scenario, more than two choices are needed for each question.

If malicious experts or non-experts communicate and predetermine the answers to questions, they can instigate collusion attacks to introduce semantic biases systematically. However, in most crowdsourcing operations, malicious workers with motivation to instigate collusion attacks are not common.

In summary, if questions have more than two choices, the randomization strategy is quite effective if malicious workers do not collude.

7 CONCLUSION

In this paper, we target an important class of answer aggregation problems for multiple-choice questions in which majority voting fails. We propose the concept of hyper questions based on the assumption that the experts are more likely to give correct answers to all of the single questions included in a hyper question, whereas non-experts do not. This allows experts to reach consensus more easily than non-experts, and thus strengthens their influences in the answer aggregation methods. We incorporate the concept of hyper questions into the existing aggregation methods and propose Hyper-MV, Hyper-GLAD, and Hyper-DARE. The results of our experiments show that the proposed methods performed better than the corresponding existing methods in the cases where only a small number of experts are available.

Although the proposed methods exhibit good performance, they have several drawbacks. First, their performance deteriorates if the missing rate of answers is large. Second, their performance is slightly sensitive to the size of the hyper questions (k) and the selection of good k is important to take full advantage of these methods. These issues will be addressed in future work.

ACKNOWLEDGMENTS

This research was supported by JSPS KAKENHI Grant Number 15H01704 and JST ACT-I Grant Number JPMJPR16UL, Japan.

REFERENCES

[1] Y. Bachrach, T. Minka, J. Guiver, and T. Graepel. 2012. How to Grade a Test Without Knowing the Answers: A Bayesian Graphical Model for Adaptive Crowdsourcing and Aptitude Testing. In *Proceedings of the 29th International Coference on International Conference on Machine Learning (ICML '12)*. 819–826.

[2] A. P. Dawid and A. M. Skene. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28, 1 (1979).

[3] J. S. Downs, M. B. Holbrook, S. Sheng, and L. F. Cranor. 2010. Are Your Participants Gaming the System?: Screening Mechanical Turk Workers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. 2399–2402.

[4] P. G. Ipeirotis and E. Gabrilovich. 2014. Quizz: Targeted Crowdsourcing with a Billion (Potential) Users. In *Proceedings of the 23rd International Conference on World Wide Web (WWW '14)*. 143–154.

[5] P. G. Ipeirotis, F. Provost, and J. Wang. 2010. Quality Management on Amazon Mechanical Turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation (HCOMP '10)*. 64–67.

[6] D. R. Karger, S. Oh, and D. Shah. 2011. Iterative Learning for Reliable Crowdsourcing Systems. In *Proceedings of the 24th International Conference on Neural Information Processing Systems (NIPS '11)*. 1953–1961.

[7] G. Kazai, J. Kamps, and N. Milic-Frayling. 2012. The Face of Quality in Crowdsourcing Relevance Labels: Demographics, Personality and Labeling Accuracy. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM '12)*. 2583–2586.

[8] H. C. Kim and Z. Ghahramani. 2012. Bayesian classifier combination. In *Artificial Intelligence and Statistics (AISTATS '12)*. 619–627.

[9] H. W. Li, B. Zhao, and A. Fuxman. 2014. The Wisdom of Minority: Discovering and Targeting the Right Group of Workers for Crowdsourcing. In *Proceedings of the 23rd International Conference on World Wide Web (WWW '14)*. 165–176.

[10] Q. Liu, J. Peng, and A. Ihler. 2012. Variational Inference for Crowdsourcing. In *Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS '12)*. 692–700.

[11] F. L. Ma, Y. L. Li, Q. Li, M. H. Qiu, J. Gao, S. Zhi, L. Su, B. Zhao, H. Ji, and J. W. Han. 2015. FaitCrowd: Fine Grained Truth Discovery for Crowdsourced Data Aggregation. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '15)*. 745–754.

[12] J. M. Mortensen, M. A. Musen, and N. F. Noy. 2013. Crowdsourcing the verification of relationships in biomedical ontologies. In *AMIA Annual Symposium Proceedings (AMIA '13)*, Vol. 2013. 1020.

[13] J. Oosterman, A. Nottamkandath, C. Dijkshoorn, A. Bozzon, G. J. Houben, and L. Aroyo. 2014. Crowdsourcing Knowledge-intensive Tasks in Cultural Heritage. In *Proceedings of the 2014 ACM Conference on Web Science (WebSci '14)*. 267–268.

[14] S. Y. Pan, K. Larson, J. Bradshaw, and E. Law. 2016. Dynamic Task Allocation Algorithm for Hiring Workers That Learn. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI '16)*. 3825–3831.

[15] D. Prelec, H. S. Seung, and J. McCoy. 2017. A solution to the single-question crowd wisdom problem. *Nature* 541, 7638 (2017), 532–535.

[16] V. C. Raykar, S. P. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy. 2010. Learning From Crowds. *J. Mach. Learn. Res.* 11 (Aug. 2010), 1297–1322.

[17] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng. 2008. Cheap and Fast—but is It Good?: Evaluating Non-expert Annotations for Natural Language Tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '08)*. 254–263.

[18] G. Tsoumakas and I. Vlahavas. 2007. Random k-Labelsets: An Ensemble Method for Multilabel Classification. In *Proceedings of the 18th European Conference on Machine Learning (ECML '07)*. 406–417.

[19] M. Venanzi, J. Guiver, G. Kazai, P. Kohli, and M. Shokouhi. 2014. Community-based Bayesian Aggregation Models for Crowdsourcing. In *Proceedings of the 23rd International Conference on World Wide Web (WWW '14)*. 155–164.

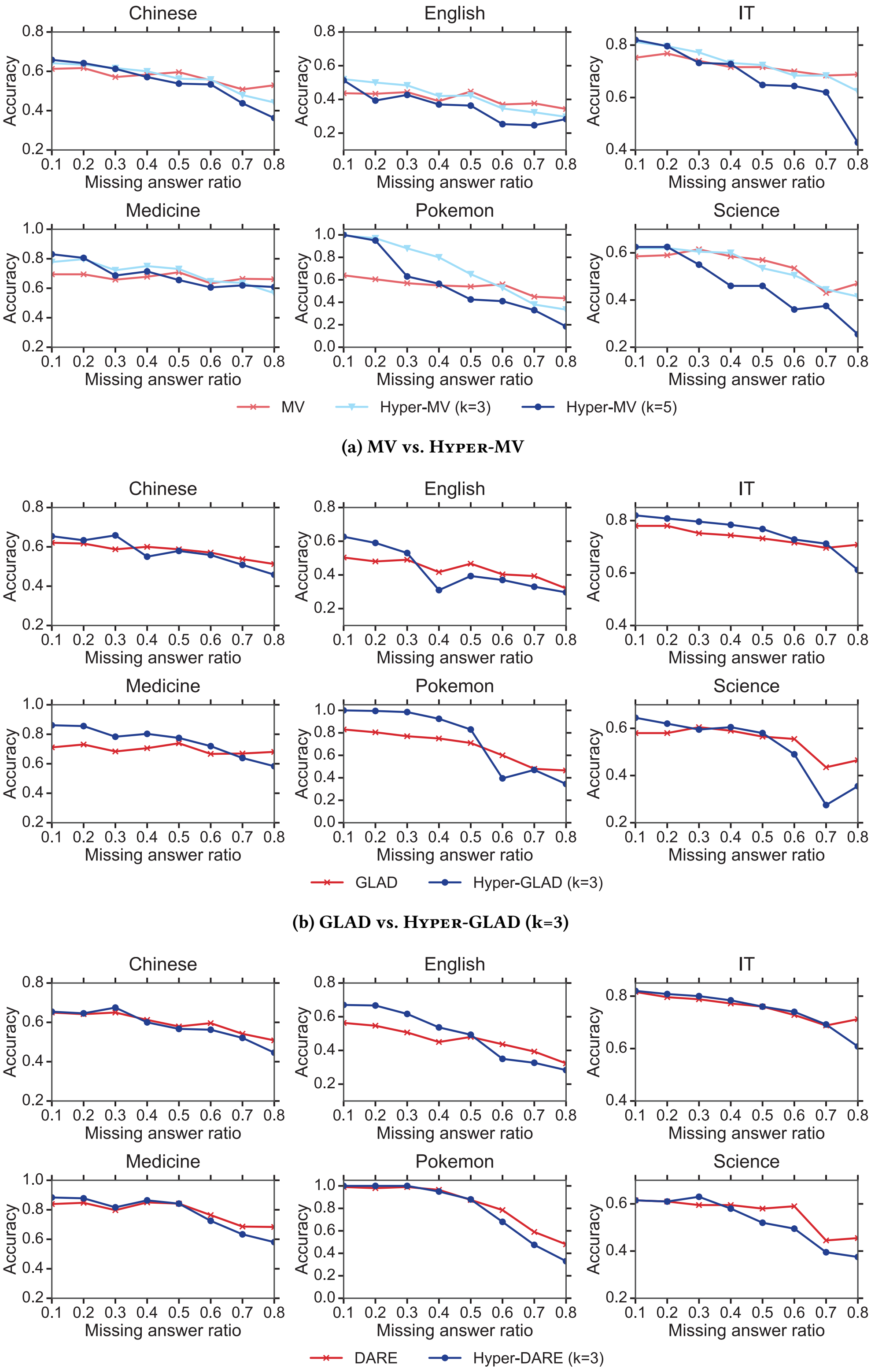
[20] F. L. Wauthier and M. I. Jordan. 2011. Bayesian Bias Mitigation for Crowdsourcing. In *Proceedings of the 24th International Conference on Neural Information Processing Systems (NIPS '11)*. 1800–1808.

[21] P. Welinder, S. Branson, S. Belongie, and P. Perona. 2010. The Multidimensional Wisdom of Crowds. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems (NIPS '10)*. 2424–2432.

[22] J. Whitehill, P. Ruvolo, T. F. Wu, J. Bergsma, and J. Movellan. 2009. Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise. In *Proceedings of the 22nd International Conference on Neural Information Processing Systems (NIPS '09)*. 2035–2043.

[23] Y. D. Zheng, G. L. Li, Y. B. Li, C. H. Shan, and R. Cheng. 2017. Truth Inference in Crowdsourcing: Is the Problem Solved? *Proc. VLDB Endow.* 10, 5 (Jan. 2017), 541–552.

[24] D. Y. Zhou, J. C. Platt, S. Basu, and Y. Mao. 2012. Learning from the Wisdom of Crowds by Minimax Entropy. In *Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS '12)*. 2195–2203.



(a) MV vs. Hyper-MV

(b) GLAD vs. Hyper-GLAD (k=3)

(c) DARE vs. Hyper-DARE (k=3)

Figure 6: Average accuracies on real datasets according to the ratio of missing answers. The average accuracies among the 10 sampling results are shown. When the missing ratio is less than or equal to 0.3, Hyper-MV ($k = 3, 5$), Hyper-GLAD ($k = 3$), and Hyper-DARE ($k = 3$) perform better than MV, GLAD, and DARE, respectively in most cases; however, their performances decline with increases in the ratio of missing answers.