Incorporating External Knowledge into Crowd Intelligence for More Specific Knowledge Acquisition

Tao Han[1], **Hailong Sun**[1], **Yangqiu Song**[2], **Yili Fang**[1], **Xudong Liu**[1]
[1]School of Computer Science and Engineering, Beihang University, Beijing, China
[2]Lane Department of CSEE, West Virginia University, Morgantown, United States
{hantao, sunhl, fangyili, liuxd}@act.buaa.edu.cn, yangqiu.song@mail.wvu.edu

Abstract

Crowdsourcing has been a helpful mechanism to leverage human intelligence to acquire useful knowledge for well defined tasks. However, when aggregating the crowd knowledge based on the currently developed voting algorithms, it often results in common knowledge that may not be expected. In this paper, we consider the problem of collecting as specific as possible knowledge via crowdsourcing. With the help of using external knowledge base such as WordNet, we incorporate the semantic relations between the alternative answers into a probabilistic model to determine which answer is more specific. We formulate the probabilistic model considering both worker's ability and task's difficulty, and solve it by expectation-maximization (EM) algorithm. Experimental results show that our approach achieved 35.88% improvement over majority voting when more specific answers are expected.

1 Introduction

Crowdsourcing [Howe, 2006] has been successfully used for leveraging human intelligence to perform tasks that computers are currently unable to do well. It has been applied to many applications such as named entity resolution [Wang *et al.*, 2013], image annotation [Russell *et al.*, 2008], audio recognition [Hwang and Lee, 2012], video annotation [Vondrick *et al.*, 2013], etc. However, when crowdsourcing is applied to knowledge acquisition, such as information extraction and image annotation, a problem of *what kind of knowledge should be acquired* arises. To our best knowledge, most aggregation algorithms for crowdsourcing results are based on majority voting or its variants. In voting approaches, aggregated answers tend to converge to common or commonsense knowledge which is usually labeled with entry-level concepts (or basic level concepts) [Waggoner and Chen, 2014]. For object recognition, it is more consistent with human that machine can recognize objects with their entry level concepts. [Ordonez *et al.*, 2013; Feng *et al.*, 2015]. However, for knowledge acquisition, more specific concepts are often preferred. On the one hand, more specific knowledge means more concrete annotations or answers to an instance or a question. On the other hand, we can easily map the specific concepts to more general concepts when having a good enough knowledge base of taxonomy. Whereas it is more difficult to instantiate a general concept to more specific concepts by a computer on the fly. For example, if we want to annotate a picture of a hummingbird and most workers label it as bird, and the voting algorithm consequently annotates it as bird, then there is no chance to acquire the knowledge of hummingbird given the fact that the decision has been made.

In this paper, we focus on how to generate more specific knowledge from crowdsourcing results. There are two major challenges for the problem. First, more specific answers are often labeled with less workers compared with common and commonsense knowledge. Therefore, it is unlikely to directly obtain such information from the voting results. Nonetheless, if we have some external knowledge showing that some concepts are subconcepts of higher level concepts, then we can derive a model to incorporate this knowledge into voting to re-weight the more specific concepts. Several knowledge bases have broad coverage of this kind of concept-subconcept relationship [Fellbaum, 1998; Lenat and Guha, 1989; Speer and Havasi, 2012; Wu *et al.*, 2012]. We employ WordNet [Fellbaum, 1998] here to serve as external knowledge.

Second, since human behaviors can contain strategies, mistakes and malevolence, how to aggregate from these unreliable multiple answers to a credible one is an important problem in crowdsourcing. Different workers may have different answering ability while different tasks may be of different difficulty for different workers. Therefore, it has been shown that incorporating worker ability and/or task difficulty into crowdsourcing decisions can significantly improve the results [Whitehill *et al.*, 2009; Salek *et al.*, 2013; Zhou *et al.*, 2012]. For more specific knowledge, the worker ability and task difficulty are more critical issues, since crowdsourcing platforms are usually not developed for any specific domain, and workers on the platforms may not be domain experts. Therefore, it will be more important to consider these two factors into the decision models. For example, we need to consider how these factors interact with the external knowledge in our case.

Given the above challenges and considerations, we propose a probabilistic model called Simplicity-ability Estimation model with External Knowledge (SEEK), in which we

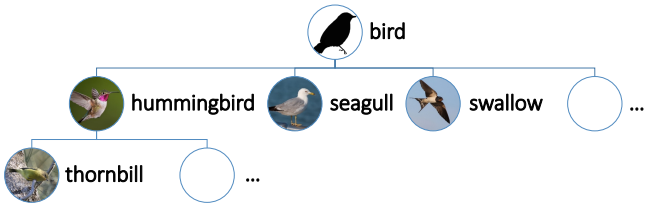

Figure 1: Example of hierarchical knowledge labels.

factorize the conditional probability of the most specific trustworthy labels with respect to task difficulty, worker ability, and the external knowledge. Here we use the term "task simplicity" instead of "difficulty" to make this factor consistent with worker ability. Then the expectation-maximization (EM) algorithm is adopted to solve this model. There have been some great studies on acquiring binary relationships to construct a taxonomy of concepts [Chilton *et al.*, 2013; Bragg *et al.*, 2013; Sun *et al.*, 2015], and also using the taxonomy to classify items based on multi-label classification [Bragg *et al.*, 2013]. Compared to their approaches that ask any binary questions in a taxonomy and intelligently choose which questions to ask by the control algorithm, our approach asks the workers input one label and decides which one among all the labels is more specific.

The contributions of this paper are summarized as follows.

- We propose a crowdsourcing problem which targets to acquire more specific knowledge from workers.
- We propose a decision making algorithm that can estimate task simplicity, user ability, and incorporate external knowledge to solve the problem.
- We conducted a set of experiments to demonstrate the effectiveness and advantages of our work in comparison with the state-of-the-art approaches.

2 Problem Formulation

In this section, we introduce our problem formulation of knowledge acquisition with crowdsourcing.

2.1 Definition of KAC

We call our problem as the knowledge acquisition with crowdsourcing (KAC) problem in general. Formally, we define the KAC problem as follow.

Definition 1 KAC Problem. *Let $D = \{d_j | j \in I_D\}$ be the unlabeled task set, $W = \{w_i | i \in I_W\}$ be the workers set, and $\Omega = \{x_k | k \in I_\Omega\}$ be the label domain set.*[1] *We denote the label set $L = \{L_1, L_2, ..., L_n\}$ where $L_j \in L$ contains labels that workers give to d_j. Namely, for $\forall d_j \in D$ we get label set $L_j = \{l_{ij} | i \in I_W, j \in I_D\}$ from workers. The problem of KAC is to find a function $f : \Omega^{|D|\times|W|} \to \Omega^{|D|}$, which generates the most specific label from all the labels provided by workers for each task.*

Let $F = \{f | f : \Omega^{|D|\times|W|} \to \Omega^{|D|}\}$ be the universal set of aggregation algorithms of KAC. Then with the well-defined

[1] I_X is an index set of set X

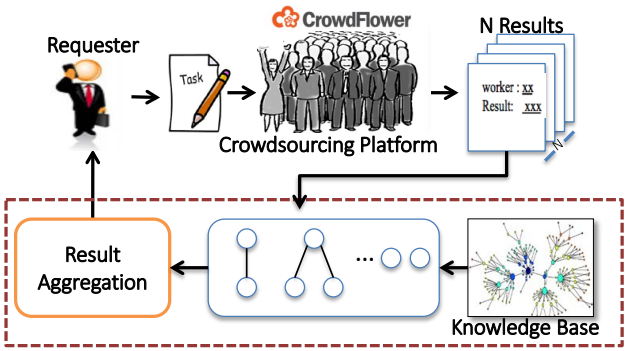

Figure 2: Workflow of crowdsourcing with external knowledge.

value function $v : F \to \mathbb{R}$ which measures quality of the algorithms, we can formulate the aggregation problem of KAC as to find a function f^*:

$$f^* = \underset{f \in F}{\operatorname{argmax}}\, v(f). \tag{1}$$

For instance, if there are 100 tasks, 10 workers, and 4 candidate labels for workers to choose, then the aggregation algorithm is to find a function with a 100×10 label matrix as input and a 100 dimensional label vector as output. Each element of the label vector is the final answer to the corresponding task in one of the 4 candidate labels.

2.2 Definition of HKAC

When the alternative answers from workers have concept-subconcept relationship with each other, we call the problem Hierarchical Knowledge Acquisition with Crowdsourcing (HKAC). In this case, the labels has the hierarchical arborescence structure as shown in Figure 1. If a label is another label's parent node, this means the concept of the first label is more general than that of the second one. Conversely, if a label is one of the child nodes of another label, this means the concept of the first label is more specific than that of the second one. The HKAC problem is to choose a label as specific as possible even when workers provide more common labels than relatively specific labels. Since voting cannot help us choose the more specific labels, we propose to use external knowledge base, i.e. WordNet, to identify the semantic relations among alternative labels. Specifically, we denote the relation function as $R : \Omega \times \Omega \to \mathbb{R}$. We introduce following notations and properties for further use.

Definition 2 Transitivity and Symmetry. *If concept c_k is a parent node of c_l, then $c_k \in \text{hypernym}(c_l)$. For $\forall c_k \in \text{hypernym}(c_l)$, we have $\text{hypernym}(c_k) \in \text{hypernym}(c_l)$. Moreover, if $c_k \in \text{hypernym}(c_l)$, then $c_l \in \text{hyponym}(c_k)$ and vice versa.*

2.3 Workflow

To incorporate the hierarchical knowledge, we propose a crowdsourcing workflow as shown in Figure 2. Different from general crowdsourcing workflows, we incorporate external knowledge to conquer the convergence of labels to

common knowledge. The steps of this workflow is listed as follows.

Step 1: A requester publishes tasks to a crowdsourcing platform, e.g. Crowdflower[2].

Step 2: The platform assigns tasks to workers according to its scheduling policies and user-specified constraints.

Step 3: For each received task, a worker provides a label which s/he believes the best to describe the object or answer the question in the corresponding task.

Step 4: After collecting all the labels from workers, we run our model with the external knowledge base to infer the aggregated result for each task. Finally, all the aggregated results are returned to the requester.

3 SEEK Model

In this section, we first show the relation function derived from external knowledge base WordNet (Section 3.1). Then we propose a naively modified majority voting algorithm to incorporate the external knowledge (Section 3.2). We further introduce a probabilistic model to let the external knowledge interact with task difficulty and worker ability (Section 3.3). Finally, we complete SEEK model and give a solution to it using EM algorithm (Section 3.4)

3.1 External Knowledge

We derive a relation function $R : \Omega \times \Omega \to \mathbb{R}$ over the label domain based on external knowledge to describe the semantic relation of labels. Specifically, $R(x_k, x_l)$ is defined as:

$$R(x_k, x_l) = \begin{cases} 1 & \text{if } x_k = x_l \\ 1 - Dist(x_k, x_l) & \text{if } x_k \in hypernym(x_l) \\ 0 & otherwise \end{cases}, \quad (2)$$

where $Dist(x_k, x_l)$ is a normalized distance between two nodes on WordNet graph. It is computed as the length of the shortest path between two nodes to their common ancestor over the length of the path from the shallower node to the root.[3] The value of function $R(x_k, x_l)$ means the scale of x_k giving a support to x_l to find the most specific labels x_l if x_k is a hypernym of x_l. Note that here we only consider two nodes being the same and single-direction hypernym relationship, and exclude other relations including hyponym.

3.2 Majority Voting with External Knowledge

In original majority voting, we evaluate each label $x_k \in \Omega_j$ based on its frequency: $\delta_{jk} = \frac{\sum_i I(l_{ij}=x_k)}{\sum_k \sum_i I(l_{ij}=x_k)}$, where I is an indicator function. In weighted majority voting, which we call Majority voting With ability Weight (MWW) algorithm, we weight each label with worker i's ability a_i: $\delta_{jk} = \frac{\sum_i a_i I(l_{ij}=x_k)}{\sum_k \sum_i a_i I(l_{ij}=x_k)}$. We can compute the work's ability simply using the aggregated label confidence $a_i = \frac{\sum_{j,k} \delta_{jk} I(l_{ij}=x_k)}{\sum_{j,k} \delta_{jk}}$.

Given the relation function, we can derive a simple Majority voting With external Knowledge (MWK) algorithm based on MWW algorithm which is shown in Algorithm 1. Given the label set L and relation matrix R, it infers an answer to each task. First, the parameters of worker ability and the score of each label are initialized to be 1 and the label frequency in the corresponding task respectively. Then, it iterates the process, during which the ability and scores update themselves using the relation matrix until convergence. Line 8 is the core of this algorithm which updates the new scores by itself $\delta_{jk}^{(n)}$ and the support $\sum_{k' \neq k} R(x_{k'}, x_k)\delta_{j,k'}^{(n)}$ which is the aggregated quantity from other equal or more general labels corresponding to the same task. The sum of the new scores corresponding to one example is not 1 because of the addition in line 8. Thus we normalize it as shown in line 9. The remaining process is the same as MWW algorithm.

Algorithm 1 Majority Voting with External Knowledge

Input: Label set $L = \{l_{ij} \in \Omega | i \in I_W, j \in I_D\}$ and relation matrix R sized of $|\Omega| \times |\Omega|$ with elements varying form 0 to 1
Output: Aggregation labels $L^T = \{l_j^T \in \Omega_j | j \in I_D\}$

1: **Initialization**:
2: Worker i's ability parameter $a_i^{(0)} = 1$
3: Score for label x_k in task j as $\delta_{jk}^{(0)} = \frac{\sum_i a_i^{(0)} I(l_{ij}=x_k)}{\sum_k \sum_i a_i^{(0)} I(l_{ij}=x_k)}$
4: **for** n = 1 to maxIter **do**
5: **if** ability error $<$ tolerance **then**
6: break
7: **end if**
8: $\delta'_{jk} = \delta_{jk}^{(n)} + \sum_{k' \neq k} R(k', k)\delta_{j,k}^{(n)}$
9: Update $\delta_{jk}^{(n+1)} = \frac{\delta'_{jk}}{\sum_k \delta'_{jk'}}$
10: Update $a_i^{(n+1)} = \frac{\sum_{j,k} \delta_{jk}^{(n+1)} I(l_{ij}=x_k)}{\sum_{j,k} \delta_{jk}^{(n+1)}}$
11: **end for**
12: $l_j^T = \text{argmax}_{x_k} \delta_{jk}$

3.3 Probabilistic Modeling

MWK considers the external knowledge and worker ability in a naive way. Now we introduce a more general and fine tuned model to incorporate both worker ability and task simplicity. From the probabilistic point of view, we regard $R(x_k, x_l)$ as a non-negative, monotonically increasing function of the probability of the label $l_{ij} = x_k$ given the aggregated label $L^T = \{l_j^T | j \in I_D\}$, namely

$$R(x_k, x_l) = g(p(l_{ij} = x_k | l_j^T = x_l)), \quad (3)$$

where $g(\cdot)$ is a monotonic function. For instance, given task j's label domain $\Omega_j = \{dog, husky, poodle\}$ and assuming worker i has normal intelligence and will do the work to her best, if we consider $L^T = \{l_j^T | j \in I_D\}$ as the perfect estimation of groundtruth labels, we have three cases as follows.

- $l_j^T = husky$ and $l_{ij} = l_j^T = husky$. As the definition in Eq. (2), we have $p(l_{ij} = husky | l_j^T = husky) = g^{-1}(1)$ where g^{-1} is the inverse function of g. It means that for a normal worker i, she will give the most specific trust label with very high probability.
- $l_j^T = husky$ and $l_{ij} = dog$. So $p(l_{ij} = dog | l_j^T = husky) = g^{-1}(1 - Dist(dog, husky))$, where for ex-

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

[3]https://rednoise.org/rita/reference/RiWordNet.html

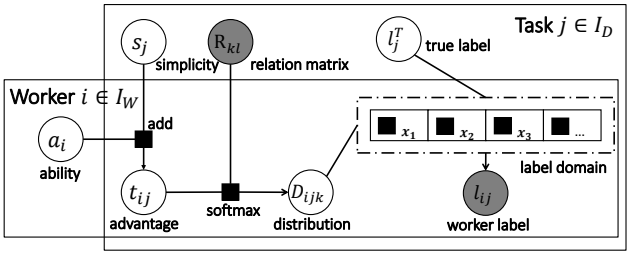

Figure 3: Factor graph of SEEK model.

ample we have $1 - Dist(dog, husky) = 0.86$) based on WordNet. It means that worker i does not know the concept of husky and she has a high probability of labeling it as "dog."

- $l_j^T = husky$ and $l_{ij} = poodle$. Here we have $p(l_{ij} = poodle|l_j^T = husky) = g^{-1}(0)$ (note that we only have positive scores when two concepts are equal or has "hypernym" relationship) which does not mean $p(l_{ij} = poodle|l_j^T = husky) = 0$ but means the probability of this case is small. It also means that worker i misunderstands the conceptual relationship of "husk" and "poodle."

Based on the discussion above and inspired by [Whitehill *et al.*, 2009], we represent the conditional probability of l_{ij} given l_j^T, a_i, s_j with a softmax function:

$$p(l_{ij}|l_j^T, a_i, s_j) = \frac{e^{(a_i+s_j)R(l_{ij},l_j^T)}}{\sum_{l\in\Omega_j} e^{(a_i+s_j)R(l,l_j^T)}}, \tag{4}$$

where $a_i \in \mathbb{R}$ is the ability parameter of worker i and $s_i \in \mathbb{R}$ is the simplicity parameter of task j. We define the advantage score $t_{ij} = a_i + s_j$ which demonstrates the advantage for worker i to figure out the task's trustworthy answer label. For a positive $t_{ij} > 0$, the larger t_{ij} is, the more possibly worker i gives the trustworthy specific label to task j. Otherwise, worker i will be regarded as venomous when $t_{ij} < 0$. Particularly when $t_{ij} = 0$, the distribution of l_{ij} turns out to be uniform which means worker i has no idea about task j and gives label randomly.

By our definition, obviously $p(l_{ij}|l_j^T, a_i, s_j)$ distributes over Ω_j if we decompose Ω by

$$\Omega = \bigcup_{j\in I_D} \Omega_j. \tag{5}$$

The size of Ω_j depends on the unique labels workers provide for task j. Thus, for a certain task, we have a small range of distribution while the total label domain can be quite large.

3.4 Inference

Based on the discussion in Section 3.3, we formally introduce the SEEK model as shown in Figure 3. We have l_{ij} being the observed labels falling in label domain Ω_j. The unobserved variables are the "perfect" label l_j^T, the ability parameter a_i, the simplicity parameter s_j, the advantage t_{ij}, and l_{ij}'s conditional probability variable D_{ijk}. In addition there is known parameter set of the relation matrices $\{R_{kl}\}$ as external knowledge. In this model, our goal is to find the posterior distribution of l_j^T and select the label l_j^T with the *maximum a posterior* estimation as the final answer to task j.

Through the addition of a_i and s_j we get variable advantage $t_{ij} = a_i + s_j$. $D_{ijk} = p(l_{ij} = x_l|l_j^T = x_k, a_i, s_j), x_k \in \Omega_j$ which is the probability in multinomial distribution. The prior distribution of l_j^T is a uniform discrete distribution over label domain Ω_j. D_{ijk} and l_j^T determines the distribution of l_{ij} which is observable.

For simplicity, we ignore the prior of a_i and s_j and we use EM algorithm to obtain maximum likelihood estimates of the parameters of a_i and s_j similar to [Whitehill *et al.*, 2009].

Expectation Step: Let $L_j = \{l_{ij}|i \in I_W\}$, $a = \{a_i\}$ and $s = \{s_j\}$. Then for $\forall j \in I_D$, we compute $p(l_j^T|L, a, s)$ as:

$$p(l_j^T|L, a, s) = \frac{p(l_j^T)\prod_i p(l_{ij}|l_j^T, a_i, s_j)}{\sum_{l_j^T\in\Omega_j} p(l_j^T)\prod_i p(l_{ij}|l_j^T, a_i, s_j)}. \tag{6}$$

Maximization Step: Let $L = \{l_{ij}|i \in I_W, j \in I_D\}$, and $L^T = \{l_j^T|j \in I_D\}$. We compute the standard auxiliary function Q:

$$\begin{aligned}
&Q(a^{old}, s^{old}, a, s)\\
=&E\left[\ln p(L, L^T|a, s)\right]\\
=&\sum_j E\left[\ln p(l_j^T)\right] + \sum_{ij} E\left[\ln p(l_{ij}|l_j^T, a_i, s_j)\right]\\
=&Const + \sum_{ij}\sum_{l_j^T\in\Omega_j} p(l_j^T|L, a^{old}, s^{old}) \ln p(l_{ij}|l_j^T, a_i, s_j).
\end{aligned} \tag{7}$$

We use the old parameters a^{old} and s^{old} to update new a and s via gradient ascent by

$$(a, s) = \underset{(a,s)}{\operatorname{argmax}}\, Q(a^{old}, s^{old}, a, s). \tag{8}$$

The derivation detail is omitted due to the lack of space. We will provide the details and the code upon the publication of this paper. The EM algorithm is summarized in Algorithm 2.

4 Evaluation

In this section, we report the evaluation results of the proposed SEEK model in terms of correctness and effectiveness. We first introduce how to generate the data we used in Section 4.1. Then we show the results based on the data we have, and compare different approaches in Section 4.2.

4.1 Data Preparation

We used the images shown in LEVAN (learn everything about anything) project [Divvala *et al.*, 2014] which provides many categories of images in different granularities of concepts. The concepts we used were chosen from the following set of top-level concepts {*bird, dog, cat, crow, horse, sheep*}. We crawled the images in different concepts and filtered out the images with dead URLs and finally obtained 631 unambiguous images for experiments.

Algorithm 2 EM Algorithm for SEEK Model

Input: Label matrix $L = \{l_{ij} \in \Omega | i \in I_W, j \in I_D\}$ and relation matrix R sized of $|\Omega| \times |\Omega|$ with elements varying form 0 to 1
Output: aggregation labels $L^T = \{l_j^T \in \Omega_j | j \in I_D\}$

Initialization:
worker i's ability parameter $a_i = 1$ v
task j's simplicity parameter $s_j = 0$
for n = 1 to maxIter **do**
  if sum of ability and simplicity errors $<$ tolerance **then**
    break
  end if
  E step:
  compute $p(l_j^T | L, a, s)$
  M step:
  update a, s by $\max_{a,s} E[\ln p(L, L^T | a, s)]$
end for
$l_j^T = \text{argmax}_{l_j^T} p(l_j^T | L, a, s)$

We followed the workflow shown in Section 2.3 based on Crowdflower, and ensured that the quality of labels by employing level 3 worker which is the maximum level of worker in the platform. We gave a brief instruction for workers to provide as specific labels as possible. For each task, workers were asked to fill a textbox with the label they gave to the image.

Originally, we planned to present the candidate label set with the corresponding concepts in WordNet on Crowdflower. However, Crowdflower does not support to dynamically extract concepts from WordNet, we have to ask workers input labels in a textbox. Hence, after we retrieved these 631 tasks and their 6,310 labels where 10 labels for each task and corrected misspelling manually, we checked the labels with WordNet and retained the labels which can be found in WordNet.

For evaluation, the "groundtruth" is not the right category provided by LEVAN but the best one that contains the most specific knowledge for each image. Moreover, the original LEVAN's annotation of the most specific category is not good enough. Thus, we fixed the groundtruth of labels l_j^T from task j's label domain Ω_j manually.[4] Each task was labeled by two colleagues in our lab and only the labels agreed by both of them were retained as ground truth. Then 344 tasks remained, and in which there were 142 tasks whose label domains contained only one label that means no aggregation is needed. Thus, we further filtered out these 142 tasks out of 344 tasks, and had finally 202 tasks for evaluation. During our labeling process, we found that the challenge of determining the "groundtruth" lies in the difficulty to distinguish very conceptually similar labels. For example, example conflict cases like crow and raven, eagle and hawk etc. Moreover, Ω_j may contain two labels which describe different objects in the same image that we cannot tell which is more specific.

Among the selected 202 tasks, there are 1789 labels annotated by 154 workers and the number of unique labels is 92, which is considerably large compared to other crowdsourcing tagging tasks. The partial distribution of these labels is shown in Figure 4. We sorted the unique labels in descent order of the frequency of original workers' labels, which is indicated as "Total" in the figure. We also show the "groundtruth" labels fixed by us in the same figure. Then comparing the distribution the labels of "Total" and "Groundtruth," we can see that the labels with high frequencies in original workers' labeling results are mostly common knowledge. Contrarily, the labels in groundtruth set are more specific labels.

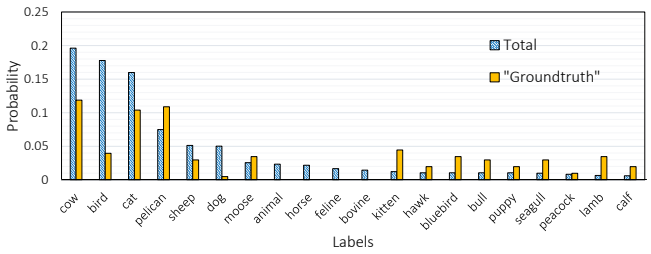

Figure 4: Partial distribution of labels.

4.2 Comparison Results

We implemented six algorithms for comparison: our SEEK algorithm, Majority Voting (MV), Majority voting With ability Weight (MWW), Majority voting With external Knowledge (MWK), Zhou's minimax entropy method (Zhou) [Zhou *et al.*, 2012] and "get another label" [Sheng *et al.*, 2008; Ipeirotis *et al.*, 2010] which is based on Dawid and Skene's method (DS) [Dawid and Skene, 1979]. Among these algorithms, SEEK and MWK incorporate external knowledge. MWK uses the knowledge in a naive way, while SEEK "learns" the parameters of a_i and s_j respectively.

The precisions of all the algorithms are shown in Table 1. Since in our overall problem, we have a much larger set of unique labels, the problem is more difficult than the problems that have been evaluated in previous works [Sheng *et al.*, 2008; Ipeirotis *et al.*, 2010; Zhou *et al.*, 2012]. We can see from Table 1 that precisions of Zhou and DS are comparable to that of MV and MWW, since they are essentially the same category of algorithms. The difference among them is how to incorporate the worker ability and task simplicity. However, it seems for our problem, the difference of how to evaluate these parameters does not affect the final results too much. Since they do not consider the specificity of the labels, purely estimating the worker ability and task simplicity may even hurt the result when the model is too complex. MWW also considers the weights to enhance the influence of the majority labels through the worker "ability." Because of the scarcity of data, the way MWW estimating the ability does not affect the results at all and MWW and MV result in the exact same accuracy. Finally we can see that SEEK's precision is 61.88% and it is 35.88% improvement compared to majority voting. It is interesting to see that MWK is also significantly better than majority voting. This means that for our problem, incorporating external knowledge may be more useful than incorporating the worker ability and task simplicity. Nonetheless, the way to estimate the worker ability and task simplicity and the way to interact with knowledge also help, which results in SEEK being better than MWK.

[4] We have released our data on https://github.com/maifulmax/IJCAI16-SEEK.git.

Table 1: Precision of different algorithms on our data.

Methods	Precision
Zhou	43.07%
DS	42.57%
MV	45.54%
MWW	45.54%
MWK	55.94%
SEEK	61.88%

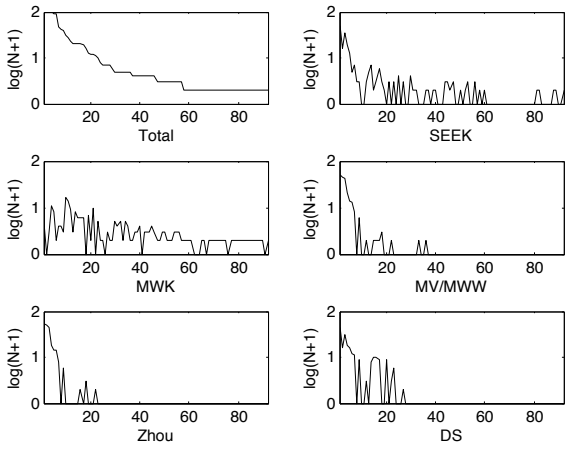

Figure 5: Log frequencies of resulting labels.

We finally report the resulting label distribution in Figures 5 and 6. In Figure 5, we compare the distribution of different algorithms separately, where the horizontal axis represents the same labels shown in Figure 4. Due to the space limit, we only show the IDs of labels instead of the labels themselves. We can see the distributions of MV/MWW, Zhou, and DS concentrate mostly on high frequency labels, which shows that they tend to vote for common labels among all the data. For SEEK and MWK, they have longer tailed distribution compared to MV/MWW. However, because of the scarcity of the data, it seems the estimation of the total label distribution is still not perfect. We also show a more detailed partial distribution in Figure 6, where we only compare MV, SEEK, and the "groundtruth" labels. We can see that for the low frequency labels like "hawk," "bluebird," and "seagull," SEEK's results are closer to the "groundtruth."

5 Conclusions and Discussion

In this paper, we identify a new problem of acquiring more specific knowledge based on crowdsourcing. We propose a novel probabilistic model that can leverage the knowledge in external knowledge bases such as WordNet. In the probabilistic model, we automatically learn the worker ability and task simplicity to customize the algorithm to fit the data. We show that using the external knowledge can achieve great improvement over voting-like methods, and learning the worker ability and task simplicity also helps improve the performance compared with naive weighting for the worker ability. Therefore, we can conclude that for the problem of acquiring more specific knowledge using crowdsourcing, both external knowledge and crowdsourcing specific parameters (e.g. worker ability and task simplicity) are important.

One remaining problem is that when we designed the crowdsourcing tasks, the workers cannot see the external knowledge base. We presume that if we can show workers with the knowledge base or if the workers can interact with the knowledge base, the final result may be better than the current one. Another problem is that majority voting still suffers from the scarcity of the data. In our problem, we have a lot of unique labels compared to previous crowdsourcing tasks, and each task may be more difficult than traditional crowdsourcing problems (if we compare common concepts with more specific concepts). Therefore, each task may need more workers to vote for a good result. Thus, if we allow more workers to label the same task, the majority voting results may also be improved. However, in this case, the cost of crowdsourcing also increases. Previously crowdsourcing has been proven to be more useful for simpler tasks. This work can be regarded as one of the first attempts that try to work on more difficult problems by combining both crowdsourcing and traditional knowledge bases.

Acknowledgments

This work was supported partly by China 973 program (2014CB340304, 2015CB358700) and partly by NSFC program (61421003). We thank Prof. Jinpeng Huai for his valuable support and contributions to this work. The authors would thank the anonymous reviewers for the helpful comments and suggestions to improve this paper.

References

[Bragg *et al.*, 2013] Jonathan Bragg, Mausam, and Daniel S. Weld. Crowdsourcing multi-label classification for taxonomy creation. In *AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*, 2013.

[Chilton *et al.*, 2013] Lydia B Chilton, Greg Little, Darren Edge, Daniel S Weld, and James A Landay. Cascade: Crowdsourcing taxonomy creation. In *SIGCHI Conference on Human Factors in Computing Systems*, pages 1999–2008. ACM, 2013.

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

[Divvala *et al.*, 2014] Santosh K Divvala, Alireza Farhadi, and Carlos Guestrin. Learning everything about anything: Webly-supervised visual concept learning. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3270–3277, 2014.

[Fellbaum, 1998] Christiane Fellbaum, editor. *WordNet: an electronic lexical database*. MIT Press, 1998.

[Feng *et al.*, 2015] Song Feng, Sujith Ravi, Ravi Kumar, Polina Kuznetsova, Wei Liu, Alexander C Berg, Tamara L

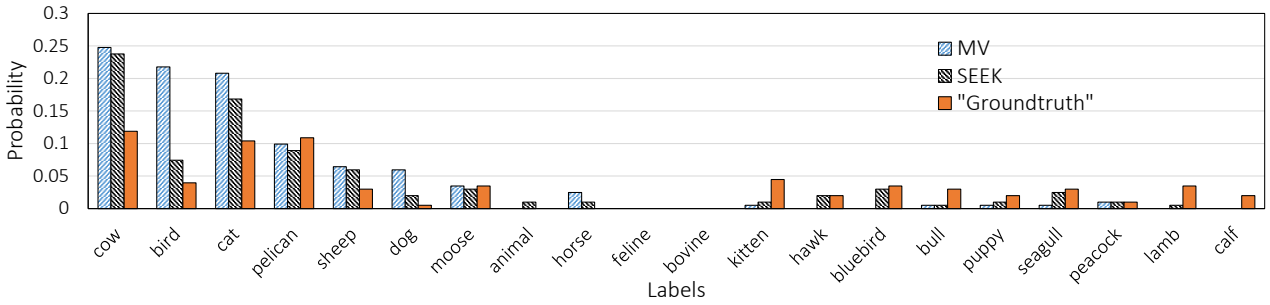

Figure 6: Partial distributions of resulting labels.

Berg, and Yejin Choi. Refer-to-as relations as semantic knowledge. In *AAAI*, 2015.

[Howe, 2006] Jeff Howe. The rise of crowdsourcing. *Wired magazine*, 14(6):1–4, 2006.

[Hwang and Lee, 2012] Kyuwoong Hwang and Soo-Young Lee. Environmental audio scene and activity recognition through mobile-based crowdsourcing. *IEEE Transactions on Consumer Electronics*, 58(2):700–705, 2012.

[Ipeirotis *et al.*, 2010] Panagiotis G Ipeirotis, Foster Provost, and Jing Wang. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 64–67, 2010.

[Lenat and Guha, 1989] Douglas B. Lenat and R. V. Guha. *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*. Addison-Wesley, 1989.

[Ordonez *et al.*, 2013] Vicente Ordonez, Jia Deng, Yejin Choi, Alexander C Berg, and Tamara Berg. From large scale image categorization to entry-level categories. In *ICCV*, pages 2768–2775, 2013.

[Russell *et al.*, 2008] Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1-3):157–173, 2008.

[Salek *et al.*, 2013] Mahyar Salek, Yoram Bachrach, and Peter Key. Hotspotting-a probabilistic graphical model for image object localization through crowdsourcing. In *AAAI*, 2013.

[Sheng *et al.*, 2008] Victor S Sheng, Foster Provost, and Panagiotis G Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *KDD*, pages 614–622, 2008.

[Speer and Havasi, 2012] Robert Speer and Catherine Havasi. Representing general relational knowledge in conceptnet 5. In *LREC*, pages 3679–3686, 2012.

[Sun *et al.*, 2015] Yuyin Sun, Adish Singla, Dieter Fox, and Andreas Krause. Building hierarchies of concepts via crowdsourcing. In *IJCAI*, pages 844–851, 2015.

[Vondrick *et al.*, 2013] Carl Vondrick, Donald Patterson, and Deva Ramanan. Efficiently scaling up crowdsourced video annotation. *International Journal of Computer Vision*, 101(1):184–204, 2013.

[Waggoner and Chen, 2014] Bo Waggoner and Yiling Chen. Output agreement mechanisms and common knowledge. In *Second AAAI Conference on Human Computation and Crowdsourcing*, 2014.

[Wang *et al.*, 2013] Jiannan Wang, Guoliang Li, Tim Kraska, Michael J Franklin, and Jianhua Feng. Leveraging transitive relations for crowdsourced joins. In *SIGMOD*, pages 229–240. ACM, 2013.

[Whitehill *et al.*, 2009] Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier R Movellan, and Paul L Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NIPS*, pages 2035–2043, 2009.

[Wu *et al.*, 2012] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Qili Zhu. Probase: a probabilistic taxonomy for text understanding. In *SIGMOD*, pages 481–492, 2012.

[Zhou *et al.*, 2012] Dengyong Zhou, Sumit Basu, Yi Mao, and John C Platt. Learning from the wisdom of crowds by minimax entropy. In *NIPS*, pages 2195–2203, 2012.