The Thesis Committee for Aashish Sheshadri
certifies that this is the approved version of the following thesis:

A Collaborative Approach to IR Evaluation

APPROVED BY

SUPERVISING COMMITTEE:

____
Kristen Grauman, Supervisor

____
Matthew Lease, Supervisor

A Collaborative Approach to IR Evaluation

by

Aashish Sheshadri, B.E.

Thesis

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Computer Science

The University of Texas at Austin

May 2014

Acknowledgments

I consider myself lucky and privileged to have Professor Matthew Lease as my advisor. His belief in my abilities has always been more than I have had for myself. His support, guidance and motivation is what made this work possible. I am sincerely thankful and forever grateful to him.

I would like to thank Professor Kristen Grauman for her valuable comments on this thesis and for being an inspiration to me.

I would like to thank Ivan Oropeza for collaborating with me, his support both as a friend and a collaborator was invaluable. I would also like to thank my labmates Hyunjoon Jung, Haofeng Zhou and Donna Vakharia for their support, advice and making this a joyful experience.

Finally, to my family and friends who have been there for me during my highs and lows, there is little I can say to truly thank them. This work exists because of my parents sacrifice, unconditional love and unabated support.

Aashish Sheshadri

The University of Texas at Austin
May 2014

A Collaborative Approach to IR Evaluation

Aashish Sheshadri, M.S.Comp.Sci.
The University of Texas at Austin, 2014

Supervisors: Kristen Grauman
Matthew Lease

In this thesis we investigate two main problems: 1) inferring consensus from disparate inputs to improve quality of crowd contributed data; and 2) developing a reliable crowd-aided IR evaluation framework.

With regard to the first contribution, while many statistical label aggregation methods have been proposed, little comparative benchmarking has occurred in the community making it difficult to determine the state-of-the-art in consensus or to quantify novelty and progress, leaving modern systems to adopt simple control strategies. To aid the progress of statistical consensus and make state-of-the-art methods accessible, we develop a benchmarking framework in SQUARE[1], an open source shared task framework including benchmark datasets, defined tasks, standard metrics, and reference implementations with empirical results for several popular methods. Through the development of SQUARE we propose a crowd sim-

[1] ir.ischool.utexas.edu/square

ulation model that emulates real crowd environments to enable rapid and reliable experimentation of collaborative methods with different crowd contributions. We apply the findings of the benchmark to develop reliable crowd contributed test collections for IR evaluation.

As our second contribution, we describe a collaborative model for distributing relevance judging tasks between trusted assessors and crowd judges. Based on prior work's hypothesis of judging disagreements on borderline documents, we train a logistic regression model to predict assessor disagreement, prioritizing judging tasks by expected disagreement. Judgments are generated from different crowd models and intelligently aggregated. Given a priority queue, a judging budget, and a ratio for expert vs. crowd judging costs, critical judging tasks are assigned to trusted assessors with the crowd supplying remaining judgments. Results on two TREC datasets show significant judging burden can be confidently shifted to the crowd, achieving high rank correlation and often at lower cost vs. exclusive use of trusted assessors.

Contents

Chapter 1

Introduction

1.1 Background

We first introduce and motivate the need for statistical consensus in the context of crowdsourcing and human computation to improve the quality of aggregated crowd labels. We then introduce building test collections for IR evaluation and motivate the need for developing scalable evaluation methodologies.

1.1.1 Statistical Consensus

Crowdsourcing platforms have enabled modern day systems to benefit either from the availability of an on-demand human computation resource [48, 8] or from the crowd as a scalable and parallelized annotation resource [1, 57, 61]. A quality concern is ubiquitous in the use of such a diverse resource, composed of individuals of varying quality and commitment.

Prior work has shown better task design to improve quality of crowd contribution, especially through multi-stage approaches like *find-fix-verify* [8]. However, crowd enabled data collection to improve data quality has often adopted the approach of eliciting redundant responses due to its task and domain-independent applicability. Consequently, statistical aggregation has been one of the most heavily investigated approach to improve quality of crowd contributions.

While many consensus algorithms have been proposed, relatively little benchmarking has occurred. As a consequence it has become increasing difficult to determine the current state-of-the-art in consensus. This has been further aggravated by in-domain development of consensus methods, lessening awareness of techniques across communities. Many researchers in other communities simply want to know the best consensus method to use for a given task, lack of a clear answer and reference implementations has led to predominant use of simple majority voting as the most common method in practice.

1.1.2 IR Evaluation

Relevance judgments provide the foundation for assessing Cranfield-based evaluation of IR systems [15]. While it is known that insufficient judgments can compromise evaluation [66], it has become increasingly challenging to manually judge so many documents as collection sizes have grown. Consequently, there has been tremendous interest in developing more scalable evaluation methodology. While commercial search engines infer implicit judgments from search logs [28], they reportedly still use many human editors for expert judging as well. Another direction of work has explored inferring judgments by retrieval popularity [59], though this fails to accurately distinguish strong vs. weak outlier systems. Pseudo-test collections cleverly simulating relevance judgments [2] or queries [6] show promise but have not been established as a general alternative.

Potential for crowdsourcing methods to improve cost, speed, ease, scalability, and/or diversity of judging vs. traditional use of trusted assessors has been established over several studies [1, 7, 9, 13, 16, 24, 34]. Bailey et al. [7] hint at collaborative approaches by showing the to be adept at identifying documents which are not relevant to a topic and hence can be useful to make a first pass and have experts only judge documents marked as relevant.

Another line of research has devised techniques by which reliable ranking of IR systems can be achieved using many fewer trusted judgments than with traditional pooling [4, 10, 12, 23, 45, 52]. However, the prevalence of these findings when using noisy crowd judgements is an open question. But these methods present a solution in determining the relative importance of judging documents to the end evaluation.

1.2 Contributions

In this thesis we make two main contributions. As our first contribution we present a comparative evaluation of aggregation methods through SQUARE (**S**tatistical **QU**ality **A**ssurance **R**obustness **E**valuation) in Chapter 2. SQUARE is a benchmarking framework with defined tasks, shared datasets, common metrics, and reference implementations with empirical results for a number of popular methods. The goal of the benchmark is to ease comparative analysis of consensus methods for the community to drive innovation and make state-of-the-art methods accessible. Through the benchmark datasets we learn different crowd properties to inform simulation and quantify the benefit of intelligent aggregation.

Following our general investigation of consensus across domains through the benchmark, we inform our study to build crowd aided systems in the context of IR evaluation. Specifically, we investigate building test collections for the evaluation of IR systems using the crowd.

As our second contribution, we bring together two lines of research, one investigating the applicability of crowds and the other investigating minimalistic judging, through an evaluation framework that enables collaboration between different judging resources (NIST TREC (expert) assessors and crowd workers).

Based on Lesk and Salton's hypothesis of judging disagreements on borderline documents [42], also studied by Voorhees [65], we propose a logistic regression model to predict disagreement and induce a prioritized order for judging enabling effective crowd and expert collaboration. We realistically simulate crowd judgements from observed crowd properties across the benchmark datasets considered in SQUARE.

We present our end to end IR evaluation framework, from building a test collection to evaluating systems on standard metrics in Chapter 3. Through the switchable design of our proposed framework we enable rapid experimentation across different crowd types and aggregation methods contextualized as cost, quality and speed. In the end we show resilience in benefit either in cost saving or coverage or both of the collaborative approach across crowd types when using intelligent aggregation.

Chapter 2

SQUARE: A Benchmark for Research on Computing Crowd Consensus

2.1 Introduction

Nascent human computation and crowdsourcing [50, 40, 41] is transforming data collection practices in research and industry. In this chapter, we consider the popular statistical aggregation task of *offline consensus*: given multiple noisy labels per example, how do we infer the best consensus label? Work in this chapter was published in [53] and additional benchmarking results from the participation at the MediaEval workshop can be found in [54].

While many consensus methods have been proposed, relatively little comparative benchmarking and integration of techniques has occurred. A variety of explanations can be imagined. Some researchers may use consensus methods to improve data quality for another research task with little interest in studying consensus itself. A natural siloing effect of research communities may lead researchers to develop and share new consensus methods only within those communities they participate in. This would lessen awareness of techniques from other communities, especially when research is tightly-coupled with domain-specific tasks. For whatever reason, it has become increasingly difficult to determine the current state-of-the-art in consensus, to evaluate the relative benefit of new methods, and to demonstrate progress.

In addition, relatively few reference implementations or datasets have been shared. While many researchers in other communities simply want to know the best consensus method to use for a given task, lack of a clear answer and reference implementations has led to predominant use of simple majority voting as the most common method in practice. Is this reasonable, or do we expect more sophisticated methods would deliver significantly better performance?

In a recent talk on computational biology, David Tse[63] suggested a field's progress is often driven not by new algorithms, but by well-defined challenge problems and metrics which drive innovation and enable comparative evaluation.

To ease such comparative evaluation of statistical consensus methods, we develop SQUARE[1] (**S**tatistical **QU**ality **A**ssurance **R**obustness **E**valuation), a benchmarking framework with defined tasks, shared datasets, common metrics, and reference implementations with empirical results for a number of popular methods. Public shared implementations and/or datasets are used when available, and we provide reference implementations for other methods.

We focus here on evaluating consensus methods which do not require feature representations for examples. This requires consensus to be computed purely on the basis of worker behaviors and latent example properties, excluding hybrid solutions which couple automatic classification with human computation. In addition to measuring performance across datasets of varying scale and properties, SQUARE varies

[1] `ir.ischool.utexas.edu/square`

the degree of supervision. Beyond empirical analysis, examining multiple techniques in parallel further helps us to organize and compare methods qualitatively, characterizing distinguishing traits, new variants, and potential integration opportunities. We envision SQUARE as a dynamic and evolving community resource, with new datasets and reference implementations added based on community needs and interest.

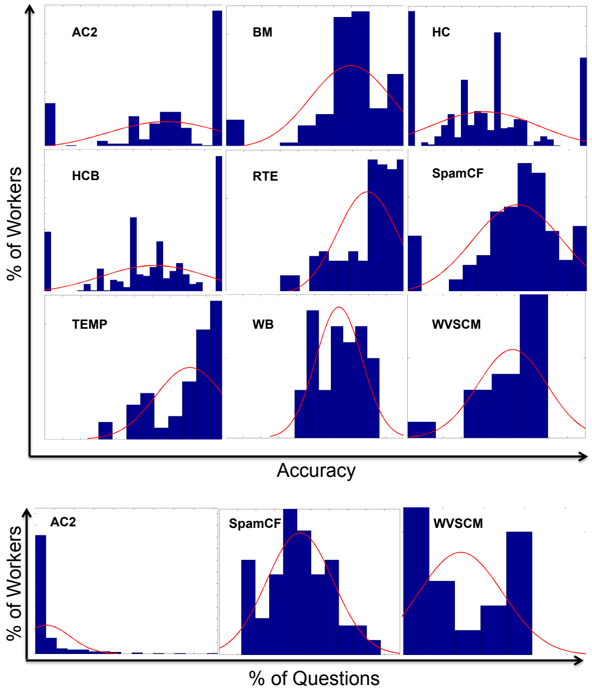

Figure 2.1: **Top**: a histogram shows the distribution of worker accuracies across nine of the datasets considered. **Bottom**: a histogram shows examples labeled per worker.

2.2 Datasets

We begin by identifying and describing a number of public datasets that are online and provide the foundation for SQUARE 1.0. An early design decision was to include only datasets containing real crowd judgments, thereby increasing validity of experimental findings. While synthetic data can also be useful for sanity checks, carefully controlled experiments, and benchmarking, relatively little synthetic data has been shared. This likely stems from its lesser perceived value and a belief that it can be easily re-generated by others (provided that the generation process is fully and aptly described, and that reproduction does not introduce errors). As Paritosh notes[49], reproducibility is both important and challenging in practice, and we posit such reproducibility is essential as a foundation for meaningful benchmarking and analysis. GLAD [71] and CUBAM [69] valuably not only provide source code for the methods evaluated, but also for generating the synthetic data used in reported experiments. Most recently, Nguyen et al. [47] present a different benchmarking study and framework based on synthetic data.

We also include only datasets with ground-truth *gold* labels for evaluation. We are agnostic here about the provenance of these gold labels and refer the reader to the source descriptions for more details. Nevertheless, the possibility of varying gold *purity* [36] should be considered in interpreting benchmark results. Not all studies creating gold labels report inter-annotator agreement statistics, and errors in gold could impact the comparative evaluation of methods considered [18].

Table 2.1 provides summary statistics for each dataset. Figure 2.1 plots a histogram of worker accuracies for nine of the datasets, above a histogram of the number of examples labeled per worker. Often, simulation based studies assume a normal distribution over worker properties which is clearly invalidated by the histograms. Further it is evident that there often exists a large group of adversarial (mis-informed or ill intentioned) workers who exhibit close to zero accuracies. While AC2 shows the oft-discussed exponential distribution of a few workers doing most of the work [22], SpamCF and WVSCM show strikingly different work distributions.

NLP Datasets. The five Natural Language Processing datasets described

Dataset	**Categories**	**Examples**	**Workers**	**Labels**	**MV *Acc.***
AC2	4	333	269	3317	88.1
BM	2	1000	83	5000	69.6
HC	3	3275	722	18479	64.9
HCB	2	3275	722	18479	64.8
RTE	2	800	164	8000	91.9
SpamCF	2	100	150	2297	66.0
TEMP	2	462	76	4620	93.9
WB	2	108	39	4212	75.9
WSD	3	177	34	1770	99.6
WVSCM	2	159	17	1221	72.3

Table 2.1: Public datasets used in the SQUARE benchmark.

below span three tasks: binary classification (BM, RTE, and TEMP), ordinal regression (AC2), and multiple choice selection (WSD).

AC2 [27] includes AMT judgments for website (ordinal) ratings $\{G, PG, R, X, B\}$.

BM [46] contains negative/positive sentiment labels $\{0, 1\}$ assigned by AMT workers to tweets.

RTE, **TEMP**, and **WSD** [57] provide AMT labels. RTE includes binary judgments for textual entailment (i.e., whether one statement implies another); expert interannotator agreement studies on gold have been reported to be 91% and 96% by prior work.

TEMP includes binary judgments for temporal ordering (i.e., whether one event follows another).

WSD includes ternary multiple choice judgments (not multi-class classification) for selecting the right sense of word given an example usage.

Other Datasets.

WVSCM [71] includes AMT binary judgments distinguishing whether or not face images smile. Gold labels were assigned by two certified experts in the Facial Action Coding System, however it is not clear is an adjudication process was adopted.

WB [69] has AMT binary judgments indicating whether or not a waterbird image shows a duck.

SpamCF [26] includes binary AMT judgments about whether or not an AMT HIT should be considered a "spam" task, according to their criteria; Gold labels were manually judged on the same criteria.

HC [11, 62] has AMT ordinal graded relevance judgments for pairs of search queries and Web pages: *not relevant*, *relevant*, and *highly-relevant*. **HCB** conflates relevant classes to produce only binary labels [29, 30]. Gold labels were assigned by trusted NIST assessors.

2.3 Models & Algorithms

Many models and estimation/inference algorithms have been proposed for *offline* consensus. Algorithms predominantly vary by modeling assumptions and complexity [44], as well as degree of supervision. Since many workers label only a few items, more complex models are particularly susceptible to the usual risks of poor estimation and over-fitting when learning from sparse data. To limit scope, we currently exclude *online* methods involving data collection, as well as methods performing *spammer* detection and removal. We also exclude consideration of *ordinal regression* methods [39], though multi-class classification methods are applicable (if not ideal). Finally, we do not consider open-ended tasks beyond multiple choice [43].

While the space of proposed algorithms is vast (far beyond what space constraints permit us to cite, describe formally, or evaluate), we consider a variety of well-known methods which provide a representative baseline of current practice. In particular, we include models which vary from ignoring worker behavior entirely, modeling worker behavior irrespective of the example, and modeling varying worker behavior as a function of example properties.

We briefly summarize and discuss each method below. Complementing empirical analysis presented in Section 2.4, our conceptual review of methods below emphasizes relationships between them, distinguishing traits, and possible variants.

2.3.1 Majority Voting (MV)

MV represents the simplest, oft-applied consensus method which often performs remarkably well in practice.

MV assumes high quality workers are in the majority and operate independently, and it does not model either worker behavior or the annotation process. It is completely task-independent with no estimation required, provides lightening-fast inference, and trivially generalizes from binary classification to multi-class classification and multiple-choice. However, this simplicity *may* come at the cost of lower label quality.

While many alternative tie-breaking strategies might be used (e.g., using an informative class prior), our formulation follows the usual practice of unbiased, random tie-breaking. Similarly, while MV assumes high quality workers dominate, a lightly-supervised variant (not reported) could detect helpful vs. adversarial workers, filtering the latter out, or with binary labeling, exploit anti-correlated labels by simply "flipping" them [38].

2.3.2 ZenCrowd (ZC)

A natural extension to MV is to weight worker responses intelligently, e.g., by the worker's corresponding reliability/accuracy. Demartini et al. [21] do so, using Expectation Maximization (EM) to simultaneously estimate labels and worker reliability. Their approach appears to be derived from first principles rather than earlier EM consensus methods [20, 56], or [57]'s passing mention of such a simplified model. Like MV, ZC makes simplifying assumptions of workers acting independently and without modeling varying worker behavior as a function of each example's true class assignment. The modeling of one parameter per worker is more complex than MV but simpler than estimating a full confusion matrix per worker. This single parameter per worker also enables detection and handling of adversarial workers, which MV cannot do without additional light supervision. An advantage of having worker reliability as the only free parameter, besides reduced model complexity for sparse data, is that the model trivially generalizes to multi-class or multiple choice tasks with no increase in complexity (though by the same token may be less effective with increasing classes or choices).

While ZC is unsupervised as proposed, it can be fully-supervised by clamping known probability estimates during maximum-likelihood (ML) iterations, lightly-supervised by only providing an informative class prior.

2.3.3 Dawid and Skene (DS) & Naive Bayes (NB)

Dawid and Skene's [20] classic approach models a confusion matrix for each worker, using EM with class priors to simultaneously estimate labels and worker confusion matrices. Snow et al. [57] adopt the same model but assume the availability of true confusion matrices, computed from supervised data with Laplacian (add-one) smoothing. Like MV and ZC, workers are assumed to operate independently [67].

Confusion matrices let DS/NB capture differential worker error behavior as a function of each example's true class. While modeling worker reliability can enable detection of adversarial workers, it is insufficient to model bias/class-expertise. Workers can be adept at labeling specific class instances or they may be biased in their responses. Such parameterization is enabled by representing each worker with a class confusion matrix, where the main diagonal encodes expertise and the off diagonal values encode confusion. While this greater modeling power can exploit more specialized statistics, sparsity can be more problematic. Also, while confusion matrices easily generalize to the multi-class labeling task, they do not generalize to the multiple choice selection task, where available choices are independent across examples.

Like ZC, DS can be generalized to light-supervision with informed class priors. A variant estimation procedure can distinguish correctable bias vs. unrecoverable noise [67]. Whereas MV is agnostic of worker behavior, and ZC models worker behavior as irrespective of the input, DS/NB model varying worker behavior given an example's true underlying class. Moreover, whereas ZC models a single parameter per worker, DS/NB model one free parameter per class per worker.

2.3.4 GLAD

Like ZC, GLAD [71] models only a single parameter per worker (the *expertise* α), with similar tradeoffs in modeling complexity; Note that unlike DS/NB, GLAD does not model worker bias. Worker expertise α, is modeled to vary from $(-\infty, +\infty)$ instead of the more traditional range of $[0, 1]$. GLAD additionally models example difficulty $1/\beta$ for each example, capturing observed label disagreement among workers; Example difficulty is modeled to vary in the range $[0, \infty]$. Likelihood of an observed label being the true class is modeled as a sigmoid parameterized by the product of α and β, making the chosen parameter space meaningful. However, the unusual ranges of modeled parameters makes prior assignments unintuitive.

Like ZC/DS, GLAD uses unsupervised model estimation via EM, but estimation is more complex, requiring gradient ascent in each *M-step*, since label probability is modeled as a sigmoid parameterized by the product of α and β.

An extension to multi-class is described (but not found in their public implementation). Like MV and ZC, GLAD easily generalizes to multi-choice selection. Like ZC and DS, gold data may be used for supervision when available (e.g., fixing known labels in EM). Light-supervision too can be enabled by assigning informed priors from observed data.

2.3.5 Raykar 2010 (RY)

DS and NB both estimate a confusion matrix, while DS imposes a class prior and NB uses Laplacian (add-one) smoothing. Raykar et al. [51] propose a Bayesian approach to add worker specific priors for each class. In the case of binary labels, each worker is modeled to have bias toward the positive class α_i (sensitivity) and toward the negative class β_i (specificity). A Beta prior is assumed for each parameter. As with ZC, DS, and GLAD, an unsupervised EM method is derived to simultaneously estimate labels and model parameters (like GLAD, involving gradient ascent).

RY's novelty lies in using an automatic classifier to predict labels, but this classifier also requires a feature representation of examples. However, when such a representation does not exist, as here, the method falls back to maximum-a-posteriori (MAP) estimation on DS, including priors on worker bias to each class. The multi-class extension is made possible by imposing Dirichlet priors, on each worker's class bias, and the class prior itself. However, the presence of class specific parameters prevents extension to multi-choice, where the available choices are independent for each example.

2.3.6 CUBAM

Methods above model annotator noise and expertise (GLAD, ZC), annotator bias (DS,NB,ZC), and example difficulty (GLAD). Welinder et al. [69] incorporate all of these along with a normalized weight vector for each worker, where each weight indicates relevance to the worker. Like prior assignments in RY, a Bayesian approach adds priors to each parameter. Worker labels are determined by an annotator-specific threshold τ_j on the projection of the noisy/corrupted input x_i and worker specific weight vector w_j.

The worker specific vector w_j can be assumed to model worker bias or other worker specific properties, while τ_j captures the expertise of the worker. This representation is more general than the confusion matrix enabling greater modeling freedom in representing a worker. The noise in the observed vector x_i captures example difficulty.

Probability of label assignments is maximized by unsupervised MAP estimation on the parameters, performing alternating optimization on x_i (example specific) and worker-specific parameters $< w_j, \tau_j >$ using gradient ascent. Apart from label estimates, the surface defined by projection $w^T \tau_j$ enables viewing worker groupings of bias and expertise. CUBAM can generalize to multi-class classification but not multi-choice selection. No direct supervised extension is apparent.

2.4 Experimental Setup

This section describes our benchmarking setup for comparative evaluation of consensus methods (Section 2.3). We vary: 1) the dataset used and its associated task; 2) the degree of supervision.

1. Data and Task. All experiments are based upon real-world crowdsourcing datasets. We use the naming convention introduced in Section 2.2 throughout this work.

2. Degree of supervision. We evaluate unsupervised performance and 5 degrees of supervision: 10%, 20%, 50%, 80%, and 90%. In each case, we use cross-fold validation, i.e. for the 10% supervision setting, estimation uses 10% train data and is evaluated on the remaining 90%, this procedure is repeated across the other nine folds, finally, average performance across the folds is reported. We report unsupervised performance on the 10-fold cross-validation setup, using 90% of examples in each fold for estimation (*without* supervision) and report average performance. Prior assignments for unsupervised estimation assumes default generic values, see

Section 2.4.1 for additional details.

In the unsupervised setting, uninformed, task-independent hyper-parameters and class priors are unlikely to be optimal. While one might optimize these parameters by maximizing likelihood over random restarts or grid search, we do *not* attempt to do so. Instead, with *light-supervision*, we assume no examples are labeled to aid estimation, but informative priors are provided (matching the training set empirical distribution). Finally, *full-supervision* assumes gold-labeled examples are provided.

To evaluate ZC, RY, DS and GLAD methods under full-supervision, labels are predicted for all examples (without supervision) but replaced by gold labels on training examples at each EM iteration.

Evaluation metrics. Presently the benchmark includes only accuracy and F_1 metrics. While a wide variety of different metrics might be assessed to valuably measure performance under alternative use cases, a competing and important goal of any new benchmark is to simplify understanding and ease adoption. This led us to intentionally restrict consideration here to two simple and well-known metrics. Significance testing is performed using a two-tailed, non-parametric permutation test [55].

Implementations. We used existing public implementations of DS, GLAD and CUBAM algorithms. We provide open source reference implementations in SQUARE for the other methods considered: MV, NB, ZC, and RY.

2.4.1 Experimental Details of Methods

A variety of important implementation details impact our evaluation of methods. We discuss these details here.

ZC in its proposed form does not impose priors on parameters [21]. Our implementation does impose priors on both the label category distribution and worker reliabilities. A Beta prior was assumed for worker reliability, and a Dirichlet prior was imposed on label categories. In each experimental setup, the workers were assigned the same prior distribution. In the unsupervised setup, the prior distribution on worker reliability had a mean of 0.7 and a variance of 0.3 (as with RY below) and the label categories were assumed to be uniformly distributed. In the lightly-supervised and fully-supervised setups, both the worker reliability and label category prior parameters were estimated from the train split.

NB was implemented to learn each worker's full confusion matrix, with Laplacian (add-one) smoothing [57]. The algorithm was extended for multi-class using a one-vs-all approach. Since NB strictly depends upon training data, it was used only in the fully-supervised setting.

RY was implemented for binary labeling [51]. Beta priors were imposed on worker specificity, sensitivity and positive category prevalence.

When unsupervised, the worker sensitivity prior was set to have mean 0.7 and variance of 0.3 (as with ZC above), the same distribution was assumed for specificity, and the label categories were assumed to be uniformly distributed. The lightly-supervised and fully-supervised settings had the prior parameters set to compute average ML estimates for each worker from the train split. Since RY was implemented for binary labeling, results are limited to datasets with two categories.

CUBAM, DS, and GLAD. Lacking supervision, CUBAM hyper-parameters were assigned default priors from the the implementation. Only the unsupervised case was evaluated since the hyper-parameters associated with distributions modeling question transformation, worker competence cannot be inferred from the train splits used.

DS predicts labels without any priors. Under the lightly-supervised and fully-supervised settings, category (class) priors were assigned ML estimates inferred from the training fold.

GLAD is assigned uniform class label likelihood, with priors of 1 for task difficulty and 0.7 for worker expertise. Under the lightly-supervised and fully-supervised settings, class priors were set by ML estimates inferred from the training fold. Worker expertise was set as the average worker accuracy inferred from the training set, and as in the other implementations, the same prior was assigned to all workers. Finally the prior on task difficulty were set to 1.

Both CUBAM and GLAD implementations support only binary class estimation, hence results from the algorithms are reported only on datasets with binary labels.

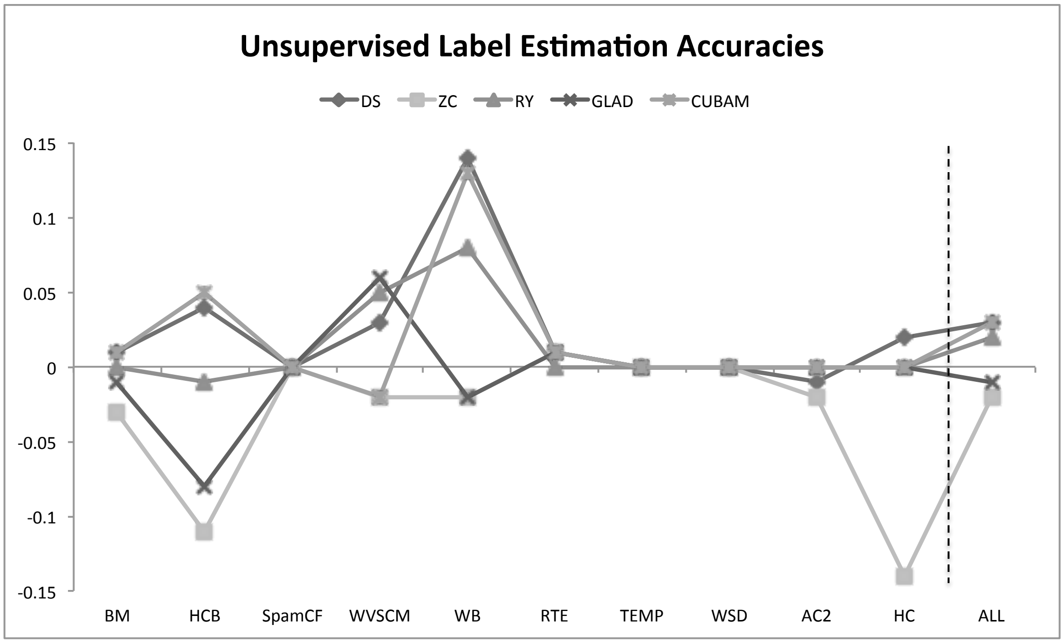

Figure 2.2: Unsupervised performance of consensus methods, as measured across seven binary labeled real datasets. Accuracy is plotted relative to a Majority Vote (MV) baseline. Average performance of methods across all datasets is shown at the right. On multiple choice WSD and multi-class AC2 and HC, results are reported only for DS and ZC.

2.5 Results

This section presents benchmarking results of methods across datasets and tasks, following the experimental setup described in Section 2.4. Statistical significance testing is limited to results in Table 2.3.

Unsupervised. Figure 2.2 plots performance of each method across each dataset, showing *relative accuracy* in comparison to the baseline accuracy of majority vote (MV). Average performance across datasets is reported both for relative accuracy to MV (Figure 2.2 far right), and for actual accuracy and F_1 in Table 2.2. Classic DS achieves top average performance for both metrics. Each method except RY and ZC also outperforms the others on at least one dataset. More strikingly, on SpamCF and TEMP datasets, methods show no improvement over baseline MV. Evaluation of the methods under the unsupervised setting, when averaged across all binary labeled datasets, showed DS to outperform the rest of the methods, both on *avg. accuracy* and F_1 score; Table 2.2 tabulates results on all the methods.

Light-supervision. Figure 2.3 plots MV relative performance for each dataset. The effect of varying supervision is shown in a separate plot for each dataset. Table 2.2 presents average results across all datasets under varying supervision. DS is seen to outperform other methods with 10%-50% supervision on *avg. accuracy* and F_1 score, but RY performs best at 90% supervision. 80% supervision has RY and DS marginally outperforming each other on *avg. accuracy* and F_1 score respectively.

Performance on each individual dataset, as observed in the unsupervised setting, did not highlight any individual method consistently performing best. Observations made earlier in the unsupervised case with regard to SpamCF and TEMP also carry-over here, with no improvement over MV for the first two.

Full-Supervision. As with previous light-supervision results, Figure 2.4 plots MV relative performance for each dataset. The effect of varying supervision is

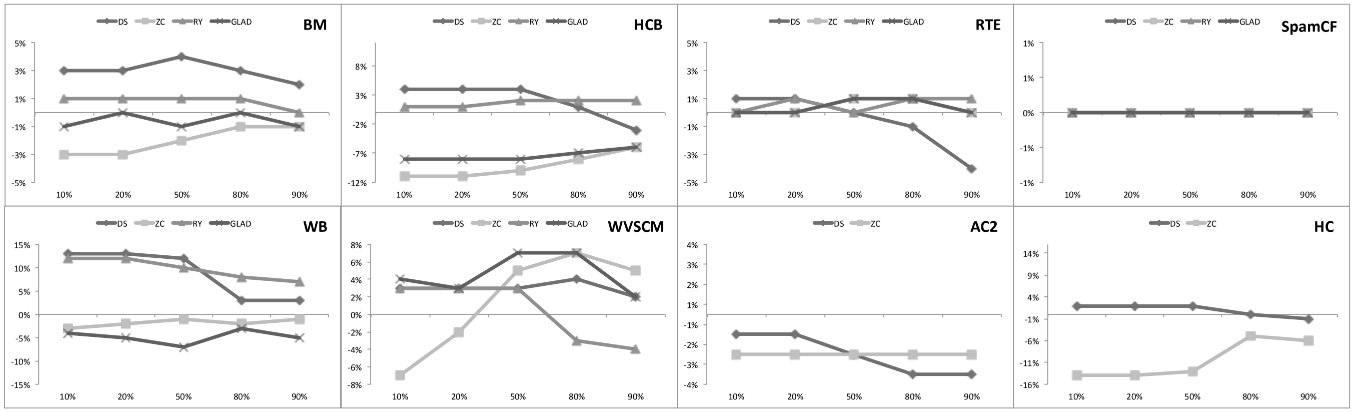

Figure 2.3: **Light-supervision.** Relative accuracy vs. baseline MV of 4 methods (DS, ZY, RY, and GLAD) across 8 crowd datasets (BM, HCB, RTE, SpamCF, WB, WVSCM, AC2, and HC) for 5 training conditions: 10%, 20%, 50%, 80%, and 90%. For multi-class AC2 and HC datasets, only multi-class methods DS and ZY are shown. Note the y-axis scale varies across plots to show dataset-dependent relative differences. Section 2.4's *Degree of supervision* provides details regarding supervision.

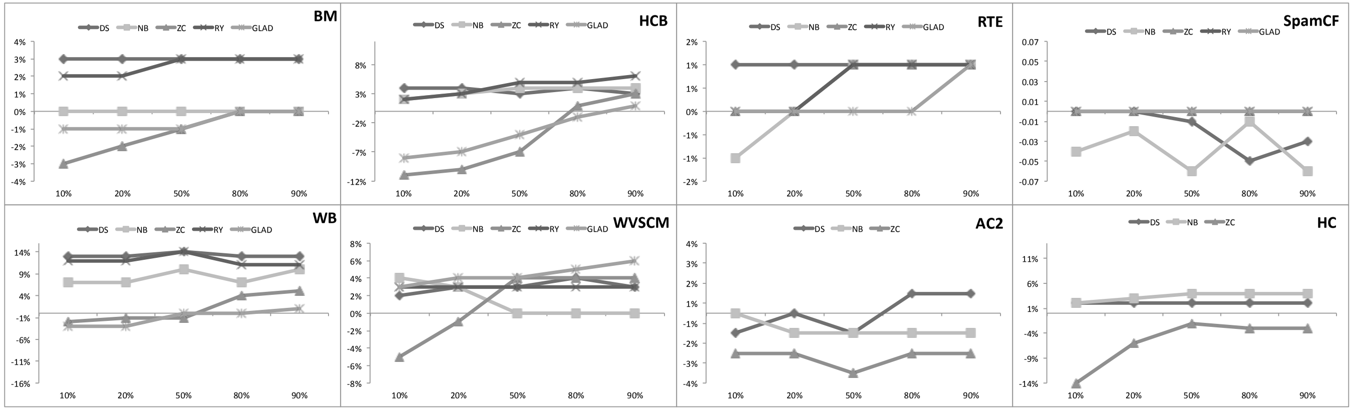

Figure 2.4: **Full-supervision.** Relative accuracy vs. baseline MV with full-supervision. See Figure 2.3 caption for further detail.

shown in a separate plot for each dataset. Table 2.2 presents average results across all datasets under varying supervision.

RY outperforms other methods with 50% or more supervision, contrasting earlier results where DS was consistently best. Note that DS outperformed the other methods for 10% and 20% supervision, but bettered RY only slightly. While NB was expected to outperform other methods with increasing supervision, DS and RY were seen to perform better.

Performance on individual datasets follows the same trend as in the averaged results, with the exception of WVSCM, where GLAD was superior. As with no supervision and light-supervision, TEMP shows similar trends, though MV outperformed DS and NB on SpamCF.

			Light-Supervision					Full-Supevision					
Method	Metric	No Supervision	10%	20%	50%	80%	90%	10%	20%	50%	80%	90%	Count
MV	*Acc*	79.2	79.2	79.2	79.2	79.3	79.3	79.2	79.2	79.2	79.3	79.3	0
	F_1	77.5	77.5	77.5	77.2	78.0	78.1	77.5	77.5	77.2	78.0	78.1	0
ZC	*Acc*	77.2	76.3	77.1	78.4	78.9	78.9	76.8	77.6	78.7	80.4	80.8	0
	F_1	76.4	74.2	75.7	76.8	77.7	77.7	75.4	76.1	77.0	79.2	79.6	0
GLAD	*Acc*	78.7	78.1	78.0	78.2	78.9	78.0	78.3	78.5	79.2	79.8	80.3	0
	F_1	77.3	76.8	76.7	77.0	78.6	77.6	76.9	77.1	77.6	79.0	79.5	0
NB	*Acc*	-	-	-	-	-	-	80.3	80.7	80.5	80.7	80.5	0
	F_1	-	-	-	-	-	-	79.1	79.0	78.5	78.5	78.9	0
DS	*Acc*	**82.2**	**82.3**	**82.2**	**82.0**	80.4	79.5	**82.2**	**82.2**	82.1	81.8	81.9	6
	F_1	$\underline{80.2}$	$\underline{80.2}$	$\underline{80.0}$	$\underline{79.4}$	$\underline{78.9}$	77.9	$\underline{80.1}$	$\underline{80.0}$	79.6	79.2	79.9	7
RY	*Acc*	80.9	81.6	81.6	81.5	**80.5**	**80.1**	81.9	82.0	**82.5**	**82.3**	**82.3**	5
	F_1	79.1	79.6	79.5	79.2	78.8	$\underline{78.8}$	79.8	79.9	$\underline{79.9}$	$\underline{80.4}$	$\underline{80.4}$	4
CUBAM	*Acc*	81.5	-	-	-	-	-	-	-	-	-	-	0
	F_1	79.8	-	-	-	-	-	-	-	-	-	-	0

Table 2.2: **Results on unmodified crowd datasets.** Accuracy and F_1 results when averaged over all seven binary datasets (BM, HCB, RTE, SpamCF, TEMP, WB, and WVSCM) for varying supervision *type* (none, light, and full) and *amount* (10%, 20%, 50%, 80%, and 90%). Maximum values for each metric across methods in each column are bolded (Accuracy) and underlined (F_1). As a simple summary measure, the final column counts the number of result columns (out of 11) in which a given method achieves the maximum value for each metric. Results of statistical significance testing (50% condition only) appear in Table 2.3.

Discussion. CUBAM, with relatively weaker assumptions, was expected to perform best. This was seen on HCB, one of the noisier datasets considered (see Figure 2.1 for its worker accuracy histogram). However, on SpamCF, a dataset with a similar noise profile to HCB, all methods perform comparably to MV. A possible explanation is that SpamCF is far smaller than HCB, challenging estimation. On the flip side, on TEMP and RTE datasets, where workers are mostly accurate, MV appears sufficient, with more complex models providing little or no improvement.

Across experimental setups, GLAD consistently performed best on WVSCM but was outperformed on other datasets. ZC performed similarly, and both model accuracy while bias is ignored. This highlights the usual value of using available domain knowledge and tuning hyper-parameters intelligently. Of course, increasingly complex models make estimation more difficult, and beyond the estimation challenge, performance is also ultimately limited by modeling capability. For datasets in which its sufficient to model worker accuracies (i.e., there exists a close to optimal positive worker weight configuration), GLAD and ZC perform well with informed priors or supervision. But they appear to be less robust on datasets with biased or adversarial workers, where methods with weak assumptions like CUBAM appear to thrive. The consistent performance of RY, across datasets, when priors were well informed or when further consolidated with minimal gold standard, suggests sufficiency in model complexity to generalize over most of the real datasets considered. Consistent performance of DS, which is similar to RY (except for the inclusion of worker priors) further corroborates this analysis.

2.6 Conclusion and Discussion

One of the motivations of SQUARE was to determine the state-of-the-art in offline consensus. While we did not find the one best method across datasets and task objectives, we observed in our benchmark tests that MV was often outperformed by some other method. More importantly the fact that each method was seen to

Dataset	Metric	Best Method-Types	Best Methods
BM	Acc	**5f**, 5l, 6lf, 7u	5-7
	F_1	**5f**, 5l, 6lf, 7u	5-7
HCB	Acc	**6f**, 5u, 7u	5-7
	F_1	**4f**, 5ulf, 6lf, 7u	4-7
RTE	Acc	**4f**, 2ulf, 3ul, 5uf, 6ulf	2-6
	F_1	**4f**, 2ulf, 3ul, 5uf, 6ulf	2-6
$SpamCF$	Acc	**7u**	7
	F_1	**7u**	7
$TEMP$	Acc	**6l**, 1u, 2ulf, 3ulf, 6u, 7u	1-3,6,7
	F_1	**6l**, 1u, 2ulf, 3ulf, 6u, 7u	1-3,6,7
WB	Acc	**4f**, 5ulf, 6lf, 7u	4-7
	F_1	**4f**, 5ulf, 6lf, 7u	4-7
$WVSCM$	Acc	**3l**, 3uf, 2ulf, 5ulf, 6ulf	2,3,5,6
	F_1	**3l**, 3uf, 2ulf, 5ulf, 6ulf	2,3,5,6

Table 2.3: **Statistical significance.** For each (unmodified) binary dataset (BM, HCB, RTE, SpamCF, TEMP, WB, and WVSCM) and quality metric (Accuracy and F_1), we report all (tied) methods achieving maximum quality according to statistical significance tests (Section 2.4). Methods are indicated by number (1=MV, 2=ZC, 3=GLAD, 4=NB, 5=DS, 6=RY, and 7=CUBAM) and supervision *type* by letter (u=none, l=light, and f=full). For each dataset-metric condition, the top scoring method-type pair is shown first in bold, followed by all tied method-type pairs according to significance tests. Given space constraints, statistical significance is reported only for the 50% supervision *amount* condition. The final column ignores supervision *type* distinctions.

outperform every other method in some condition seems to validate the need both for producing a diversity of approaches, and for multi-dataset testing in making stronger claims of improvement and generalizable performance.

We also observed method sensitivity to hyper-parameter assignments, validated by the failure to observe consistent improvement with increasing light-supervision. While investigation of more powerful models should certainly continue, we must also remain mindful of varying data conditions. Intuitively, models with few assumptions have more difficulty modeling quirks in worker behavior, over-estimating or under-estimating worker capability. However, in modeling worker bias, the classic DS and its extension RY (which effectively just adds priors on parameters) performed remarkably well across our tests. The benefit from modeling the annotation process was not observed across datasets. Better recognizing such cases through benchmarking can help us to better direct future work to specific conditions with greater opportunity for empirical improvement.

Qualitative comparison of techniques helped us to characterize distinguishing traits, new variants, and integration opportunities. Like other open source benchmarks, we envision SQUARE as dynamic and continually evolving, with new tasks, datasets, and reference implementations being added based on community needs and interest. In an independent and parallel effort, [47] recently released another open source benchmark, based on synthetic data, which implements or integrates a subset of methods found in SQUARE plus ITER [32] and ELICE [35].

Chapter 3

Collaborative Evaluation

3.1 Introduction

In Chapter 2, we investigated statistical consensus methods to leverage quality in crowdsourced data. In doing so we built the SQUARE benchmark enabling access to consensus algorithms representative of current practice and access to different crowd and task types. In this chapter we propose a collaborative judging model that combines judgments from expert NIST assessors and crowd workers. To enable rapid experimentation across different crowd types we develop a realistic crowd simulation model which emulates crowd types investigated in SQUARE . We further extend findings on the utility of consensus methods through experimentation on the simulated data. This is joint work with Ivan Oropeza. Ivan Oropeza contributed in implementing the evaluation component of the developed framework.

Relevance judgments provide the foundation for assessing Cranfield-based evaluation of IR systems [15]. While it is known that insufficient judgments can compromise evaluation [66], it has become increasingly challenging to manually judge so many documents as collection sizes have grown. Consequently, there has been tremendous interest in developing more scalable evaluation methodology. While commercial search engines infer implicit judgments from search logs [28], they reportedly still use many human editors for expert judging as well. Another direction of work has explored inferring judgments by retrieval popularity [59], though this fails to accurately distinguish strong vs. weak outlier systems. Pseudo-test collections cleverly simulating relevance judgments [2] or queries [6] show promise but have not been established as a general alternative.

One fruitful line of research has devised techniques by which reliable ranking of IR systems can be achieved using many fewer trusted judgments than with traditional pooling [4, 10, 12, 23, 45, 52]. Another stream of research has investigated potential for crowdsourcing methods to improve cost, speed, ease, scalability, and/or diversity of judging vs. traditional use of trusted assessors [1, 7, 9, 13, 16, 24, 34]. In this chapter, we bring together both lines of research into a combined experimental framework, enabling us to investigate both approaches in parallel and their interacting effects.

Figure 3.1 shows our system architecture. Given a set of document retrieval lists from IR systems to be evaluated, we first prioritize retrieved {topic,document} pairs into a judging queue [25]. Inspired by recent work of Webber et al. [68], we enable this prioritization by ordering documents by probability of disagreement as predicted by our method described in Section 3.3.

Following findings from the SQUARE benchmark developed in Chapter 2, Section 3.4 describes a method for inducing a realistic crowd model conforming to statistical properties of each crowd dataset (Section 2.2) and compares the benchmarked consensus algorithms for aggregating judgments from each crowd model.

Given a judging budget, a ratio for expert vs. crowd costs, and a crowd quality model, critical judging tasks are assigned to trusted assessors (as determined by the prioritization component), with remaining tasks delegated to the crowd. Each retrieval list is then scored for a given a ranking metric based upon our expert-crowd

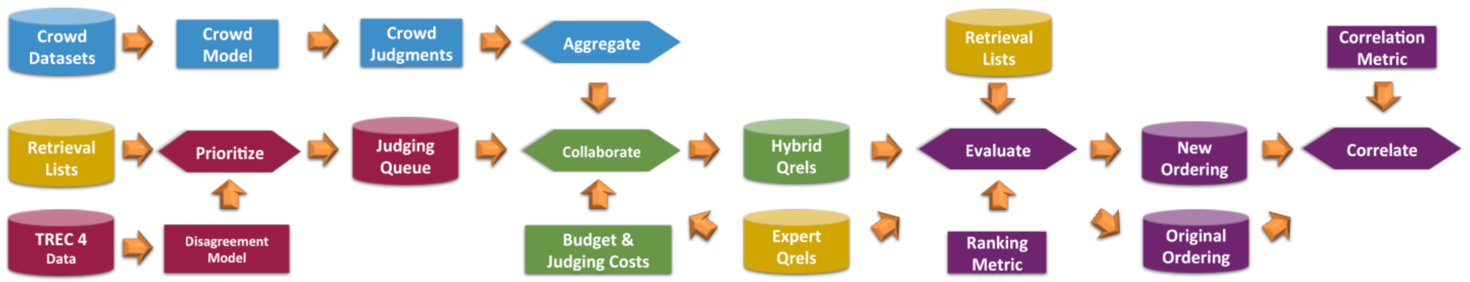

Figure 3.1: **Overall experimental framework**. Given document retrieval lists from a set of IR systems to be evaluated, we prioritize retrieved of {topic,document} pairs into a judging queue. Based on Lesk and Salton's hypothesis of judging disagreements on borderline documents [42], we build a model to predict assessor disagreement using TREC 4 retrieval lists and alternate assessments. Judgment tasks are then prioritized by expected assessor disagreement. Given a judging budget, a cost function for expert vs. crowd judging, and the judging queue, the top n judgment tasks (where disagreement is expected) are assigned to trusted assessors, with remaining judgment tasks delegated to the crowd. Next, given a set of real-world crowdsourcing datasets exhibiting varying crowd behaviors, we extract statistical properties to model varying realistic crowds. Crowd judgments are then generated from each crowd model and aggregated according to different statistical algorithms. For a given ranking metric, the combined set of expert-crowd hybrid judgments are then used to evaluate each IR system's retrieval list and rank the IR systems accordingly. Finally, we measure rank correlation of system rankings according to our hybrid qrels vs. system rankings according to original NIST qrels, according to some specified rank correlation metric.

hybrid judgments, with IR systems then ranked accordingly. Finally, we measure correlation between system rankings according to our hybrid qrels vs. use of original NIST qrels.

Section 3.2 introduces our flexible and extensible open source system architecture we develop in which system components by design are easily varied and replaced. One can vary how: 1) judgments are statically or dynamically prioritized; 2) how crowd judgments are generated or collected; 3) how crowd judgments are aggregated; 4) judging budget; 5) expert vs. crowd cost function; 6) ranking metric; 7) rank correlation metric; and 8) test collection being studied.

Finally, our main experiments evaluate TREC 6 and WebTrack 2011 participating systems using expert-crowd collaborative judging with varying budget. In comparison to accepted system rankings according to full NIST assessment, we measure the correlation of alternative system rankings according to Kendall's Tau and Yilmaz et al.'s `APCorr` [72]. Figure 3.7 and Figure 3.8 shows correlation achieved for WebTrack 2011 and TREC 6 respectively, varying crowd quality, distribution of expert vs. crowd judging, and the relative cost ratio between groups.

Results show significant judging burden can be confidently and scalably shifted to the crowd while maintaining high rank correlation, and often doing so at lower cost vs. traditional practice of using only trusted assessors. However, results suggest high sensitivity to quality of the overall crowd [37]. With less accurate workers, while we can still delegate a significant portion judging burden to the crowd and maintain high rank correlation, the cost of doing so may exceed the cost of exclusively using trusted experts. In such cases, the speed, ease, scalability, and/or diversity of crowdsourcing may still recommend it, but not cost savings.

3.2 System Architecture

Our open source experimental framework is shown in **Figure 3.1** and available for download. Our design is intended to allow system components to be easily exchanged or replaced for rapid experimentation under varying conditions.

Prioritizing Judging. Given a set of document retrieval lists from IR systems to be evaluated, we first prioritize {topic,document} pairs into a priority queue for judging (Section 3.3). We investigate two static ordering methods; dynamic schemes (cf. [3, 12, 45]) could also be used to re-order the priority queue as judging progresses. Based on Lesk and Salton's hypothesis of judging disagreements on borderline documents [42], also studied by Voorhees [65], we prioritize judging tasks by expected disagreement, as predicted by a logistic regression model. Such an ordering easily extends itself to incorporating crowd judges to judge documents with small probabilities of predicted disgareement, since making relevance judgements can have valid disagreements for which adopting the judgement of the topic originator may be the right choice and crowd judging can introduce disagreements due to varying expertise which can be corrected by intelligent aggregation. In addition, we also compare to ordering documents by average rank in retrieval lists.

Modeling Crowds. Given a set of real-world crowd datasets (Section 2.2), we learn a custom crowd model for each which defines a probability distribution over some 400 worker *archetypes* (Section 3.4). On one hand, we firmly believe crowdsourcing studies should use real data collected from crowds to ensure validity and realism of findings. On the other hand, simulation studies permit free, rapid, and more controlled studies over a wider range of possible crowd conditions. Our goal in building and sharing this crowd simulator is to balance these competing needs for realism vs. range of experimentation, letting us better study the potential and limits of crowdsourcing [16] across realistic conditions.

Aggregating Crowd Judgments. Quality is leveraged from redundant relevance judgments elicited from crowd workers by applying benchmarked consensus algorithms from SQUARE (See Chapter 2). Our system architecture allows alternative aggregation schemes to be easily compared to one another (Section 3.4.3).

Distributing Judging Assignments. Inspired by Bailey et al.'s study [7], suggesting use of *Bronze* judges as pre-filters to reduce judging effort of *Gold* and *Silver* judges, our expert-crowd collaboration model assigns the most important judging tasks to trusted assessors, while delegating the burden of remaining judging to the crowd (Section 3.5). Given a priority queue over judging tasks, a budget, and a ratio of expert vs. crowd costs, we vary the relative proportion of work delegated in determining judging assignments. The output of our collaboration model is a set of collaborative expert-crowd judgments for evaluating IR systems.

Evaluating Systems and System Rankings. The objective of Cranfield-style system evaluation [15] is to reliably measure system effectiveness given a set of relevance judgments. As in many prior studies, we vary the total number of judgments used, seeking to reduce effort and cost by using fewer judgments [4, 10, 12, 23, 45, 52]. More central to this work, we compare multiple sources of relevance judgments: trusted NIST assessors vs. crowds of varying quality. While any ranking metric can be used, we focus particularly on `BPref` [10] due to its robustness for evaluating systems with incomplete judgments. We adopt Soboroff's revised `BPref` [58], which supersedes the original formulation [10].

Measuring Rank Correlation. An evaluation metric enables a total ordering of systems. The goal of measuring rank correlation is to measure how reliably we rank IR systems under our reduced, expert-crowd collaborative judgments, vs. the ranking of systems under full NIST assessment of a judging pool. We adopt both the oft-reported Kendall's Tau and Yilmaz et al.'s more recent `APCorr` [72]. Assuming it is most important to correctly order the top-n best performing systems, Kendall's Tau is oft-criticized for equally penalizing all swaps, regardless of rank position. Various corrections have been proposed, of which we adopt `APCorr`.

3.3 Prioritized Judging

Given a set of document retrieval lists from IR systems to be evaluated, we begin by prioritizing {topic,document} pairs into a priority queue for judging [25]. Based on Lesk and Salton's hypothesis of judging disagreements on borderline documents [42], also studied by Voorhees [65], we learn an extensible logistic regression model to predict assessor disagreement using overlapping assessments and retrieval lists from past TREC evaluations (Sections 3.3.1 and 3.3.2). We train the model on TREC 4 (Section 3.3.3) and evaluate prediction accuracy on TREC 6 (Section 3.3.3). Finally, we prioritize judging tasks by expected disagreement and compare to ordering documents by average rank in retrieval lists. Kendall's Tau and `APCorr` [72] rank correlation with respect to full NIST judging is reported for WebTrack 2011 and TREC 6 *ad hoc* tasks (Section 3.3.4).

3.3.1 IR System Evaluation Datasets

WebTrack 2011. The *ad hoc* task used 50 topics. Each system submitted up to 3 ranked lists of 10K documents each. Judging was limited to a pool depth of 25, formed over all 62 submissions. In total, 19,381 documents were judged for 5-point graded relevance, which we binarize. Following Voorhees and Harman's estimate of assessors making two judgments per minute [64], judging 8 hours a day would still require over 20 person days of work.

TREC 6. The *ad hoc* task used 50 topics. Each system submitted up to 3 ranked lists of 1K documents each, with 74 total submissions. Judging was limited to a pool depth of 100, using only one retrieval list per system. A total of 72,270 binary judgments were made, requiring over 75 person days of work, per Voorhees and Harman's estimate [64].

Data	Alternate Assessor	Precision	Recall
TREC 4	A	81.3	52.8
	B	81.9	61.8
TREC 6	A	64.5	43.0

Table 3.1: Assessor agreement statistics for TREC 4 [65] and TREC 6 [17] *ad hoc* tasks. Agreement is shown in each case with respect to the original NIST assessor for each topic.

3.3.2 Assessor Disagreement Datasets

TREC 4. Secondary judgments were made by two alternate assessors [65]. Based upon original judgments, 200 relevant and 200 non-relevant documents were randomly selected for secondary judging. Table 3.1 shows the precision and recall of secondary judgments vs. the original assessor.

TREC 6. U. Waterloo provided secondary judging [17]. Interactive search by reissuing queries yielded a new document set that was judged by an single alternate assessor. Table 3.1 shows precision and recall of alternate judgments.

3.3.3 Predicting Assessor Disagreement

Our approach to predicting assessor disagreement is inspired by the approach proposed by Webber et al. [68]. However we look at the consequence of disagreement in a different light. Because the primary assessor defines the topic, we treat the primary assessor as the topic authority against which secondary assessors should be compared (rather that treating all assessors as exchangeable). Voorhees [65] reported less than 3% disagreement on judgments originally judged non-relevant, also showing that unanimously relevant documents had higher average rank than other documents. Consequently, with three judges (TREC 4), we distinguish unanimous agreement of all three judges vs. any disagreement.

The current feature space is comprised of two features:

1. **Meta-AP.** Meta-AP [5] weights Average Precision (AP). Evaluating AP at depth N imposes a weight on document at rank k as $1+H_N-H_k$, where H_n is the nth harmonic number. When k is greater than N, the weight is 0. Meta-AP implicitly assigns greater weight to documents higher in retrieval order across runs. Meta-AP is computed on each retrieval list and then averaged. We use typical depth of $N = 1000$.

2. **Weighted Avg. Retrieval Score.** To counter Meta-AP's sharp drop-off, we impose a gradually varying weighting function computed by $C_1 \cdot NS_1 + C_2 \cdot NS_2 + C_3 \cdot NS_3$, where NS_1, NS_1, NS_3 are the number of systems retrieving the document at rank $0-10$, $11-100$ and $101-1000$, respectively. We use weights $C_1 = 10$, $C_2 = 5$ and $C_3 = 1$, tuned on development data.

While we do not report feature-analysis experiments, introducing the second feature substantially improved modeling accuracy. Because these features span the entire ranking, we also experimented with histogram features using bins over narrower rank position ranges, but we did not see any improvement and so omitted these features for parsimony.

For each TREC 4 topic, we learn a topic-specific disagreement model. Model parameters, including prediction threshold, were tuned over 100 random trials of a 70%-30% train-tune split of documents judged for the given topic. Erring on the side of caution, so that judging tasks are routed to trusted assessors whenever disagreement seems plausible, we use prediction threshold 0.3 to favor higher recall. Figure 3.2 shows the 12 topics retained after training, as well as the within-topic predictions results for each topic.

To make predictions on another test collection, we must match a given test topic to one of the 12 models learned from TREC 4. In practice, we predict disagreement using all 12 models, then select the model which best agrees with a prior model, a beta distribution with $\alpha = 1.4$ and $\beta = 4$ parameters, tuned over 100 trials

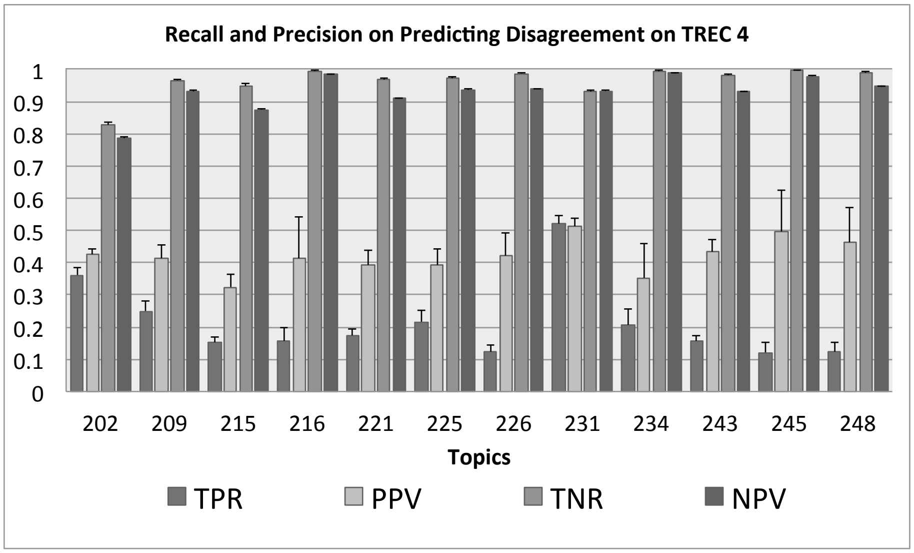

Figure 3.2: Topic-specific logistic regression model learned for each TREC 4 topic. For each of 100 trials, we make a random 70/30 train/test split of judged documents for each topic. Topics on which we fail to achieve at least 0.15 average recall are discarded; we favor recall to conservatively flag any topic on which assessor disagreement is plausible. The 12 retained, best performing models shown here will be used to predict disagreement on TREC 6 and WebTrack 2011. TPR = true positive rate (i.e., recall), PPV = positive predictive rate (i.e., precision), TNR = true negative rate (i.e., recall of negative class examples), and NPV = negative predictive value (i.e., precision of negative class predictions).

on a 70%-30% train-tune split across the 50 TREC 6 topics. We select the TREC 4 topic whose prediction minimizes Root Mean Squared Error (RMSE) vs. predictions made by the prior distribution.

Our proposed approach differs from the disagreement model developed by Webber et al. [68] in the following: 1) while they require *a priori* knowledge of the primary assessor's judgment in order to predict disagreement, we predict disagreement using only retrieval list features; 2) they limit their feature space to Meta-AP, we add an additional feature to counter the sharp drop in Meta-AP values; 3) while they train and test their model on the same topic (and the train/test division of documents is unspecified), we train our model on one test collection and test on another; and 4) unlike us they do not evaluate prediction accuracy, instead using their model to perturb the original assessor's labels in order to simulate realistic secondary judgments, we evaluate classification performance directly (Section 3.3.3), as well as measuring the evaluation benefit of prioritizing judging tasks by expected disagreement (Section 3.3.4).

Figure 3.3 shows results of predicting assessor disagreement on TREC 6. A boxplot is shown for each of 4 evaluation metrics.

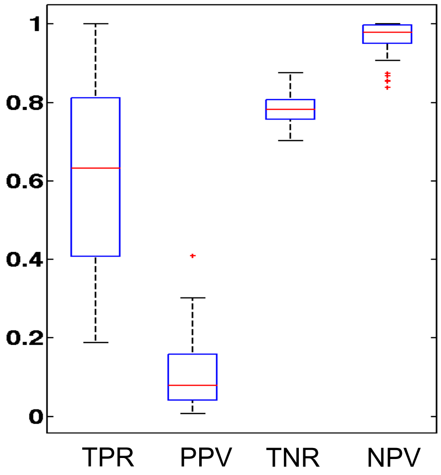

Figure 3.3: Evaluation of disagreement prediction on TREC 6 for each of 4 evaluation metrics: true positive rate (i.e., recall), positive predictive rate (i.e., precision), true negative rate (i.e., recall of negative class examples), and negative predictive value (i.e., precision of negative class predictions). Each metric's boxplot shows its median value and score distribution across topics. For non-relevant documents (TNR and NPV), prediction is most accurate and consistent across topics, with mean TNR = 0.78 and NPV = 0.96. Mean precision (PPV) was both far lower (0.11) and less consistent, due to our favoring recall over precision, and predictions for relevant documents being more difficult. Mean recall (TPR) across topics was 0.62. Recall also exhibited the highest variance across topics, due to a widely varying number of relevant documents per topic and high variance in disagreement itself, indicating some topics were far easier to judge.

3.3.4 Prioritizing Judging by Disagreement

Figure 3.4 plots Kendall's Tau and Yilmaz et al.'s `APCorr` [72] rank correlation achieved on WebTrack 2011 and TREC 6 *ad hoc* tasks. On WebTrack 2011, ordering by expected disagreement outperforms the average rank ordering consistently on Kendall's Tau and reaches the highest correlation on both correlation measures. With 32% judging, disagreement ordering achieves substantially better APCorr as well. On TREC 6, disagreement ordering shows improvement for judging 8% and above. With prior work often regarding 0.9 Kendall's Tau as acceptable correlation (cf. [65]), we see disagreement ordering achieve this using only 16% judging.

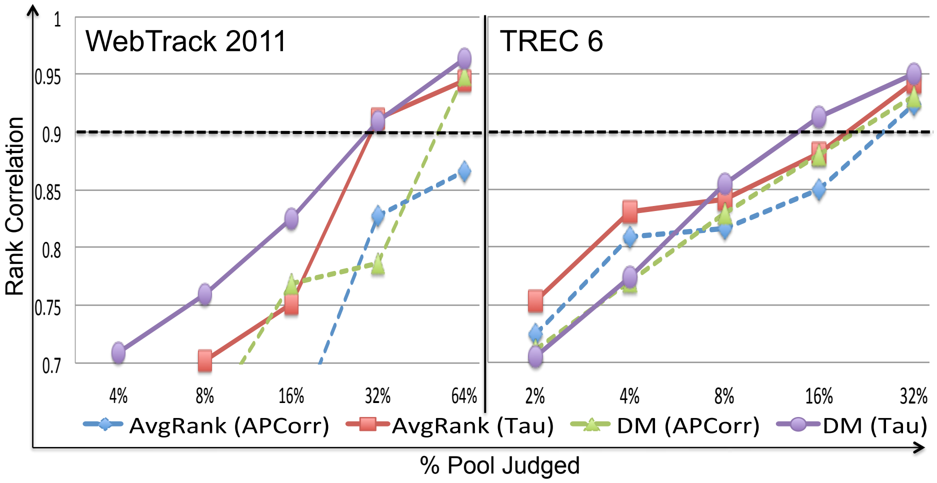

Figure 3.4: We compare prioritizing judging by the disagreement model (DM) vs. ordering documents by average rank in retrieval lists. The percentage of the original pool judged is indicated on the x-axis. TREC 6 judging percentage is varied between 2-32% by powers of 2. Since WebTrack 2011 was only judged to a depth of 25 with many fewer judgments (Section 3.3.1), judging percentage is varied here from 4-64% by powers of 2. Judgments are sampled from original NIST assessments. Participating systems are evaluated by `BPref` [10], due to its robustness with incomplete judgments. We adopt Soboroff's revised [58] formulation of `BPref`.

3.4 Judging with Crowds

A recent surge of studies have begun investigating the potential of online crowds to improve the cost, speed, ease, scalability, and/or diversity of relevance judging vs. traditional assessors [1, 7, 9, 13, 16, 24, 34]. However, with reliability of crowd data heavily dependent on quality task design (cf. [1, 9, 34]), data quality in practice can vary greatly. Different crowds may also exhibit systematic biases based on background, training, or task design [33, 60].

While we firmly believe crowdsourcing studies should use real crowd data to ensure validity and realism of findings, simulation studies remain valuable tools for free, rapid, and more controlled experimentation over a wider range of possible conditions. To balance these competing needs for realism vs. range of experimentation, we develop in this work (and share) a realistic crowd simulator which models crowds based on worker behavior statistics extracted from real-world crowd datasets. A wide diversity of real crowd behaviors are induced from the varying datasets used. Section 3.4.1 summarizes the datasets considered, while Section 3.4.2 describes our method of inducing crowd models from these datasets.

To enable benefit from crowd contributions as evident from findings in 2, it is essential to impose offline or online quality assurance methodologies [24, 34]. Section 3.4.3 measures benefit of applying statistical consensus methods from SQUARE. We also discuss how different aggregation methods impact rank correlation for IR system evaluation.

3.4.1 Crowd Datasets

We use public datasets identified for the SQUARE benchmark in Chapter 2 and an additional dataset MediaEval which is a record of fashion relevance judgements for images [54]. We only consider datasets that are applicable to our simulation framework discussed in Section 3.4.2. We follow the convention of naming datasets as introduced in Section 2.2 as BM, HCB, RTE, TEMP, WB, WVSCM and finally MediaEval. Of these, only HCB comes from the IR community, specifically the TREC 2010 Relevance Feedback track [11]. Diversity of the datasets is evidenced through their origins from natural language processing, machine vision, and multimedia research communities.

Figure 2.1 plots histograms of crowd worker accuracies and percentage of examples judged across datasets, highlighting dataset diversity. Table 3.2 shows participation, scale, and quality of each crowd dataset, as well as the quality of our crowd model's Majority Vote (MV) aggregated judgments for each dataset. It is particularly important with IR to consider class imbalance, with many fewer relevant documents and accuracy providing a less meaningful metric of label quality. Moreover, false positives (non-relevant documents erroneously judged relevant) are known to degrade evaluation reliability more than false negatives (relevant documents mislabeled as non-relevant). Table 3.2 shows that recall can be high with low precision, evidencing such false positives. We also see that some crowds perform better on the majority class, corresponding here to the easier task of judging non-relevant documents.

Crowd	**E**	**W**	**L**	***R***	***P***	***SPC***	***NPV***
BM	1000	83	5000	67.2	**96.7**	**99.6**	94.0
HCB	3275	**722**	**18479**	94.2	31.0	59.1	98.1
MediaEval	**3532**	202	1373	**98.2**	96.0	99.2	**99.6**
RTE	800	164	8000	98.0	66.1	90.2	**99.6**
TEMP	462	76	4620	94.7	63.1	89.2	98.9
WB	108	39	4212	74.7	70.1	93.8	95.0
WVSCM	159	17	1221	93.2	49.2	81.2	98.4

Table 3.2: Traits of 7 public crowd datasets used. E/W/L denote the total number of examples/workers/labels, respectively. Quality of majority vote worker labels vs. gold labels are measured by accuracy, recall, precision, specificity (recall of negative class examples), and the negative predictive value, NPV (precision of negative class predictions).

3.4.2 Modeling Crowds

To realistically simulate a worker to make highly imbalanced relevance judgments, two key properties are essential: True Positive Rate (TPR, or recall) and the True Negative Rate (TNR, or specificity). Regarding amount of work performed by each worker, prior work has consistently reported worker contribution to follow a power law distribution. Capturing this property is therefore also essential to faithfully emulate a real crowdsourcing environment. Note that while this model is rather limited in that it does not take into consideration task specific dynamics such as instructions, design, cost and worker demographics, it is general enough to represent crowd data from any system. Our goal is to enable a sufficiently abstract crowd representation from crowd data which is representative of a specific task design and worker moderation. However, we do note that failure to model example properties such as example difficulty is a shortcoming of the model.

Workers are represented in a three dimensional space defined by TPR, TNR, and PC: Percentage Contribution (Figure 3.5). TPR and TNR span [0,1] and PC spans [0,100%]. TPR and TNR for each worker are learned from worker statistics on each dataset. Similarly, PC is learned from the distribution of work performed by each worker. To enable sampling, the worker space is discretized and represented as a three-dimensional regular voxel grid. Each voxel defines a worker *archetype* in which worker properties for that archetype are modeled by the voxel's own unique multivariate Gaussian distribution. The probability distribution over voxels is learned from the frequency of representative workers matching that archetype in a given dataset.

Figure 3.5 visualizes this discretized space for two of the crowd models, HCB and MediaEval, using 10 levels each for TPR and TNR, and only 4 levels of PC (given the aforementioned power-law distribution of quantity of work performed). Each crowd model assumes this same three-dimensional representation and discretization, but varies in terms of both: 1) the probability distribution over voxels; and 2) the $10 * 10 * 4 = 400$ voxel-specific Gaussian models.

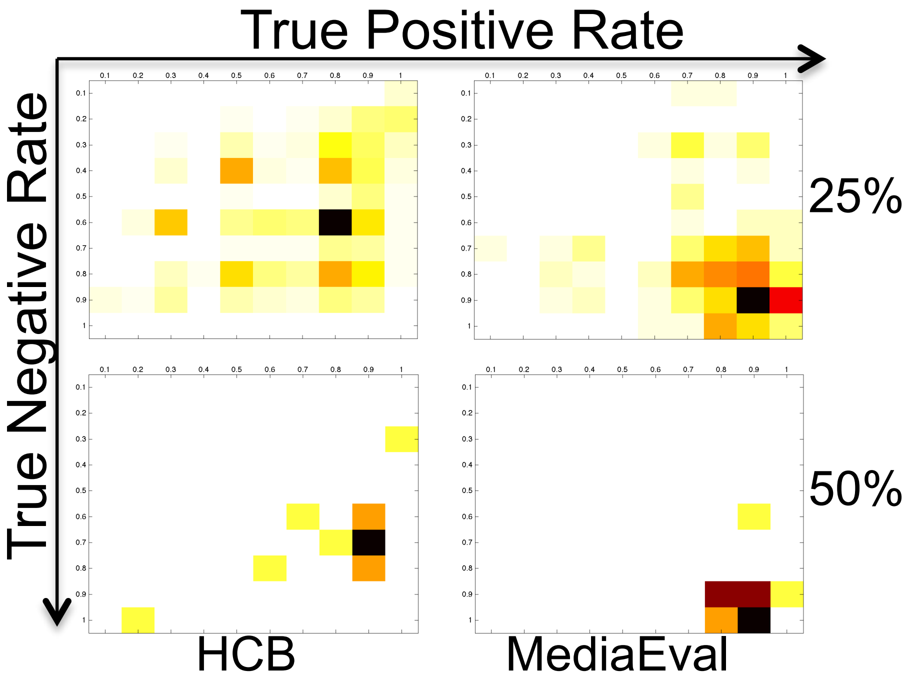

Figure 3.5: A crowd model is defined by a probability distribution over 400 worker *archetypes*, letting us generate crowd workers for each model by sampling from this distribution. A worker archetype is defined by 3 parameters: true positive rate (i.e., recall), true negative rate (i.e., specificity); and the proportion of total examples labeled, percentage contribution (PC). This 3-dimensional parameter space is evenly quantized into 10 levels each of recall (x-axis) and specificity (y-axis), and 4 PC levels (z-axis). All crowd models use the same $10*10*4 = 400$ worker archetypes in varying proportion. A crowd model for a given crowd dataset is estimated by computing a 3-dimensional histogram over observed worker statistics, i.e., the relative proportion of workers whose (recall, specificity, PC) lies in a particular bin. The top two 10x10 grids above visualize the 0-25% contribution quanta for HCB (left) and MediaEval (right). The shade of cells in each grid indicates the % of workers in the given bin, with darker shade indicating more workers. The lower two grids show the 25-50% contribution quanta.

To generate a worker, a voxel is sampled according to the crowd model's voxel distribution. Next, a worker is generated from the voxel by sampling from the voxel's unique multivariate Gaussian distribution. Each generated worker is characterized by an unique TPR, TNR and PC.

Though a traditional judging model involves an assessor judging all documents for a given topic [13], we assume a more typical crowdsourcing setup in which many workers will not judge so many documents, as determined by the PC parameter. Nonetheless, we do assume that the judging task is setup such that each worker judges only a single topic (or would finish a topic before beginning work on another).

Crowd judgments are generated by perturbing the original trusted assessor's judgment. If the document were originally deemed relevant, the crowd worker makes the same judgment with probability TPR; if the document were not relevant, then TNR is used similarly.

Typical to crowd workers tending to be liberal in assigning (possible) relevance, the TNR tended to have higher variation (See Figure 3.5). To improve aggregate judgment quality, we assume each document assigned to the crowd is judged by five unqiue workers.

3.4.3 Aggregating Crowd Judgments

Our investigation in Chapter 2 was motivated to find the best method for label aggregation, but on the contrary, findings indicated a more dataset specific performance bias. Hence, we evaluate five different aggregation methods from SQUARE (Section 3.4.3) against simple Majority Vote (MV). Unlike the evaluation procedure followed in 2.4, we limit aggregation to be unsupervised.

Table 3.3 reports recall and precision of all six consensus methods across all seven crowd models induced from the different datasets. Crowd models for each dataset are used to generate relevance judgments for WebTrack 2011, aggregated by each consensus method, and then compared against trusted NIST assessments.

For recall, MV outperforms other aggregation methods on three crowd models, however at the cost of poor precision. While no single aggregation method appears to be a clear winner across crowd models, similar DS and RY models do outperform all other methods for most crowd models. On HCB, the noisiest of the considered models, all the methods achieve dismal precision.

Seeing that HCB and MediaEval crowd models lie at opposite ends of the quality spectrum, we next apply each model to generate crowd judgments for WebTrack 2011. We compare 5 of the aggregation methods: CUBAM, GLAD, MV, Raykar, and ZC. The y-axis shows `APCorr` [72] rank correlation vs. original NIST assessments. The left-most point in each plot represents rank correlation with no trusted assessors and 32% of the pool judged by each crowd model. No difference in aggregation algorithms is observed for the HCB crowd model, while for the MediaEval crowd, ZC vs. CUBAM aggregation varies by roughly 7% `APCorr`.

3.5 Collaborative Judging

Our over-arching goal is to enable a dependable and scalable approach to test collection construction. At one extreme, judging effort might be delegated entirely to the crowd, potentially compromising on quality. At the other extreme, we have traditional practice of using only trusted assessors, with its known scalability limitations. We seek to bridge this divide through enabling effective collaboration between trusted assessors and crowd judges.

Recall Section 3.3 developed a logistic regression model for predicting assessor disagreement. Section 3.4 later discussed inconsistency observed across the many crowd models considered. As in active learning with noisy labels, asking crowd judges to assess documents having high probability of disagreement seems likely to introduce noise into estimated ranking of IR systems. Instead, we investigate assigning such judgments to trusted assessors, and delegating easier judging tasks

		AGGREGATION ALGORITHMS					
Crowd	**Metric**	**MV**	**CUBAM**	**DS**	**GLAD**	**RY**	**ZC**
BM	R	67.18	**94.46**	62.40	66.30	55.31	51.09
	P	96.67	43.82	89.06	96.14	98.92	**99.20**
HCB	R	94.17	87.01	74.12	91.38	**94.74**	93.35
	P	30.93	28.85	24.31	31.25	**35.76**	35.34
MediaEval	R	98.19	**98.42**	97.37	97.37	93.85	96.07
	P	95.98	54.89	91.03	97.19	**99.30**	99.05
RTE	R	**98.00**	94.20	89.04	96.26	91.32	91.80
	P	66.14	47.67	83.61	77.35	89.51	**91.59**
TEMP	R	**94.74**	93.06	79.38	87.39	87.17	76.31
	P	63.06	56.60	**92.20**	85.58	91.76	88.27
WB	R	74.66	**78.21**	16.44	71.43	58.70	44.57
	P	70.09	66.75	**96.65**	82.63	90.74	90.95
WVSCM	R	**93.16**	87.74	44.28	88.79	76.88	58.12
	P	49.16	50.63	57.08	61.05	**81.28**	69.22

Table 3.3: An unique crowd model is induced for each of 7 public crowd datasets [53]. Worker labels are generated according to each crowd model and aggregated under six different consensus algorithms: majority voting (MV), CUBAM [69], Dawid-Skene (DS) [20], GLAD [70], Raykar (RY) [51], and ZenCrowd (ZC) [21]. Recall and Precision of consensus crowd labels vs. gold labels are shown for each dataset.

to the crowd. Note that our experiments here assume the trusted assessor is actually the topic developer (since we are using NIST qrels, and this reflects their judging process). As such, our reported findings are conservative in that secondary assessors would likely be less reliable, in which case we would expect to see even greater relative benefit from our use of crowds than our results here indicate.

Given the judging queue ordered by expected disagreement, the top k judging tasks are assigned to trusted assessors, while the rest are distributed among crowd workers. The depth k parameter is induced by a judging budget, a ratio of expert vs. crowd costs, and desired evaluation reliability, as measured by `APCorr` [72] rank correlation with respect to the correct ranking of systems according to full NIST

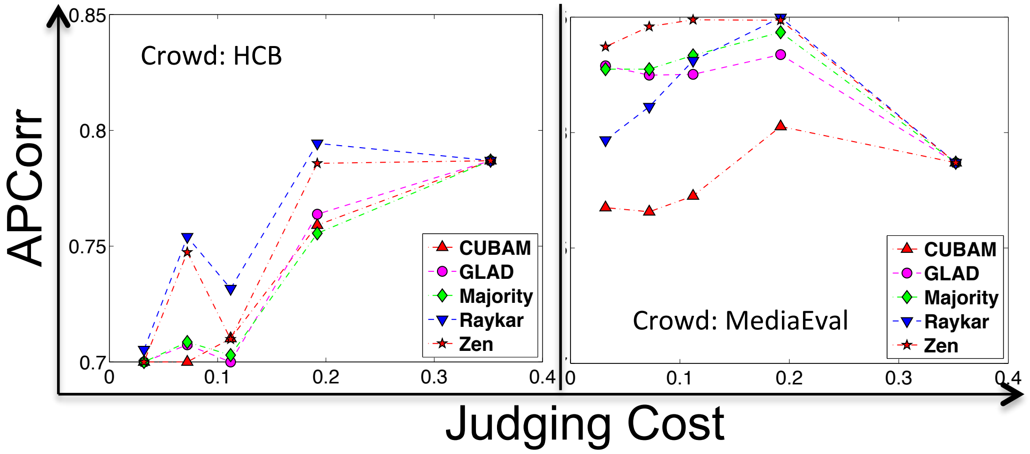

Figure 3.6: Rank correlation achieved on WebTrack 2011 using partial pool judging by a collaboration of trusted assessors and crowd judges. Assuming 5 crowd judgments per document for each document assigned to the crowd, we compare 5 alternative algorithms for aggregating crowd judgments (see Table 3.3). Given the priority queue for judging, the first 0-32% of judgments are assigned to trusted assessors, with the crowd supplying a fixed 32% additional judging. The cost ratio of 1 expert judgment vs. 5 crowd judgments is set at 10:1. The x-axis shows combined cost of collaborative judging as a fraction of original cost of having the full pool judged by trusted assessors. `APCorr` [72] rank correlation is given on the y-axis in relation to the original ordering of systems with full pool judging by NIST. Beyond the 40% cost shown at the right extent of each plot, all aggregation algorithms converge to a single line (not shown).

assessment. Systems are evaluated by BPref to reliably measure performance given incomplete judgments.

We consider two cost ratios A and B in measuring the combined cost of collaborative judging. Cost ratio A assumes a cheaper crowd ratio of 10:1 – 1 trusted judgment costing the same as 10 consensus crowd judgments (each aggregated in turn from 5 individual worker judgments). The more expensive 5:2 cost ratio B assumes 2 trusted judgments cost the same as 5 consensus crowd judgments.

To ease analysis and generalize findings, this section reports on only the two most contrasting crowd models, optimistic MediaEval and pessimistic HCB, with performance of other crowd models expected to lie between these two extremes. Section 3.5.1 reports effectiveness of our collaborative judging approach on WebTrack 2011, while Section 3.5.2 reports effectiveness on TREC 6. See Section 3.3.1 for details on test collections.

3.5.1 Collaborative Judging on WebTrack 2011

To investigate the benefit of using an intelligent aggregation method over naive MV an experiment is set up to measure APCorr across collaboration levels of 0% to 64% from an expert and a fixed 32% crowd effort. Figure 3.6 plots the result of the experiment on the two crowd models HCB and MediaEval. On HCB, the noisier crowd model, at 0% expert assistance RY measured the best correlation. However it did not outperform the rest of the methods with a large margin, this was expected as discussed in Section 3.4.3. With increasing expert collaboration, a difference between the aggregation methods is evident with other methods outperforming MV. On MediaEval, a cleaner dataset, we observe the simplest aggregation method ZC to outperform the rest with MV being competitive. This suggests that if an assessment of the participating crowd is available, this information can help choose an aggregation method. In the absence of such knowledge, consistent with findings of the SQUARE benchmark in Chapter 2, RY is measured to be dependable on noisier and clean crowds. Another interesting observation is the diminishing benefit from aggregation methods on rank correlation with expert collaboration levels excess of 32%, while there is still benefit in reduced label noise.

To validate the proposed approach of enabling coverage at a reduced cost using crowd judgments, an experiment similar in nature to that described in Section 3.3.4 is set up. However, here additional judgments are added from the crowd. The percentage of contribution from the crowd is varied over the same scale (0% to 64%). Thus the maximum coverage of the document pool using collaborative judging is

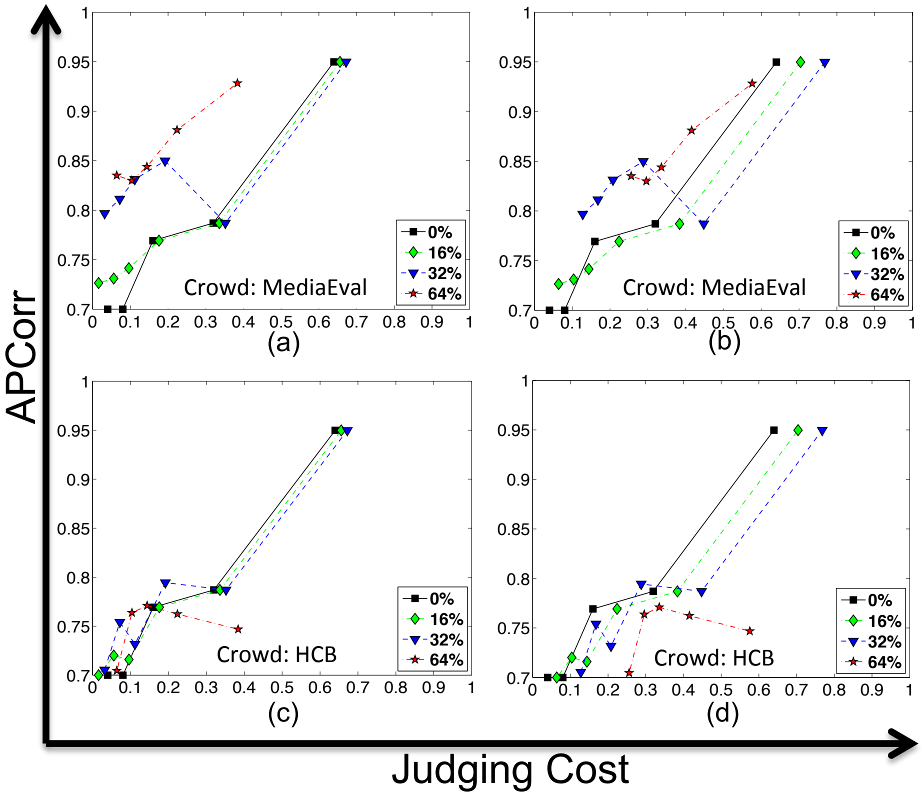

Figure 3.7: Rank correlation achieved on WebTrack 2011 using partial pool judging by a collaboration of trusted assessors and crowd judges. Documents assigned to the crowd for judging are judged independently by five workers and aggregated via RY. Given the priority queue for judging, we vary the % of trusted assessor judgments from 0-64% (by powers of 2), with the crowd supplying an additional 0-64% judgments (by powers of 2, and without exceeding 100% in total). Each plotted line corresponds to a particular % judged by the crowd, with the solid black 0% line representing traditional use of trusted assessors only. Following each line from left-to-right, markers indicate increasing increments of trusted assessor judging, from 0-64%. We omit 4% and 8% crowd lines for WebTrack 2011, which closely track 0%, and the 4% crowd line for TREC 6. The x-axis shows total judging cost incurred as a fraction of the original cost with the full pool judged by trusted assessors. Left plot assumes a liberal cost ratio of 10:1 for 1 expert judgement vs. 5 crowd judgments, while the right plot assumes a more conservative 5:2 cost ratio. The top row of plots use the optimistic MediaEval crowd model, while the bottom plots use the pessimistic HCB crowd model. Correlation with Kendall's Tau (not shown) consistently exceeds APCorr results shown, further confirming high correlation at the right extent of each plot.

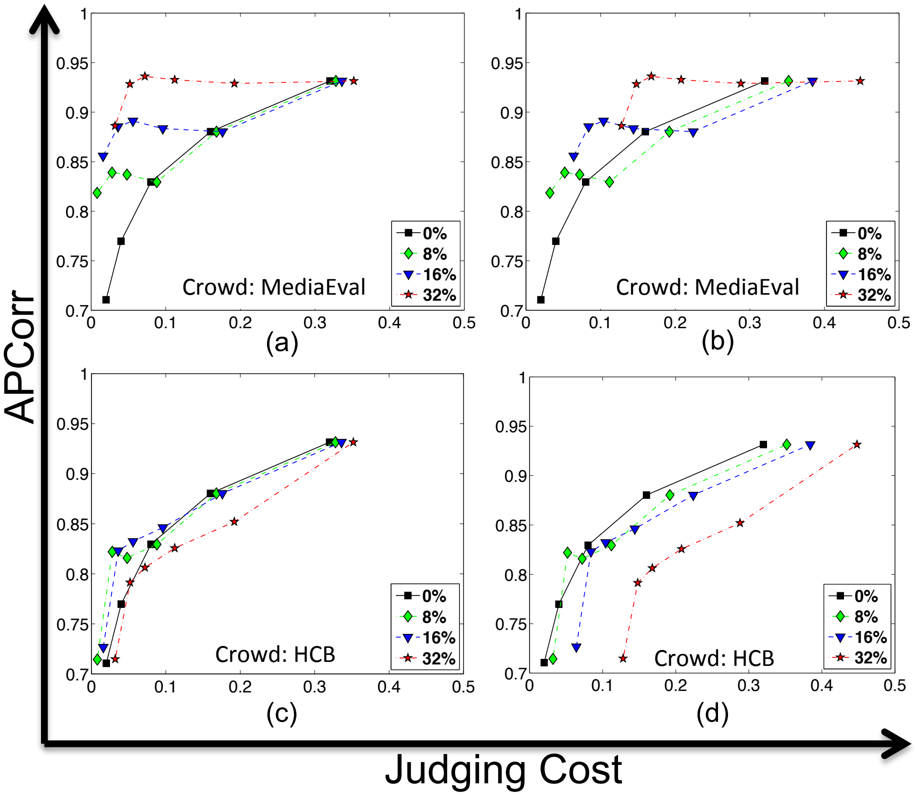

Figure 3.8: Rank correlation achieved on WebTrack 2011 using partial pool judging by a collaboration of trusted assessors and crowd judges. Refer to Figure 3.7 caption for plot details.

96%. Figure 3.7 plots rank correlation over the different collaborative efforts for the two datasets MediaEval and HCB with the two cost models A and B. As discussed earlier since RY was observed to be dependable across datasets, the figure only plots aggregation using RY. The following discusses experimental results on the two crowd models.

MediaEval. The benefit of additional judgments from this crowd type is clearly evident at all contribution levels and both cost models. A rank correlation greater than 0.9 is achieved with an expert contribution of 32% and a crowd contribution of 64%. If the cost of judging the whole document pool by an expert is considered to be 100%, the cost incurred using the collaborative approach is less than 40% on cost model A, and enables a 96% coverage. Alternatively, assuming cost model B incurs a cost of less than 60%. If the judging expense is spent solely on experts, the rank correlation measured is less than 0.8 with a coverage of only 32%. However, if cost model B is assumed, the spending on the expert allows only for a 64% coverage but enables a higher rank correlation. A non-improving rank correlation is observed with an expert contribution 16% and higher and a crowd contribution of 32%. Crowd contribution levels of 4% and 8% measure similarly on rank correlation as not using the crowd at all and hence not shown in Figure 3.7.

HCB. As earlier discussed, using only the crowd to evaluate systems resulted in poor measurements of rank correlation. Interestingly, even as little as 4% contribution from an expert enables better rank correlation across crowd contributions. Of the different crowd contributions, the contribution level of 32% consistently either performed better or equalled the performance of using just expert judgments. When assuming the cost model A, the benefit also translated to both a saving in judging resource and better coverage. However, on the more expensive cost model, the benefit was largely seen in enabling more coverage of the document pool. As in the case with the previous crowd type, a similar trend of non-improving rank correlation was observed with increasing expert contribution. Like the other crowd model, here too crowd contribution levels of 4% and 8% (omitted in Figure 3.7) compare similarly on rank correlation as not using the crowd at all. With 64% crowd contribution, increasing expert contribution measures a drop in rank correlation; This suggests the presence of critical documents lower in the prioritized order that require accurate judging, especially since the pool depth for WebTrack 2011 was only 25.

3.5.2 Collaborative Judging on TREC 6

The experiment is setup similar to that described in Section 3.3.4, however here expert judging depth is varied from 2% to 32%. As in the experiment described in Section 3.5.1, additional judgments are added from the crowd. The maximum coverage of the document pool using collaborative judging is 64%, i.e., the first 32% judged by an expert the remaining 32% judged by the crowd. Figure 3.8 plots rank correlation over the different collaborative efforts for the two datasets, MediaEval and HCB, and the two cost models; As discussed in Section 3.5.1, this figure too only plots aggregation using RY. The following discusses experimental results on the two crowd models.

MediaEval. On this crowd model, a rank correlation greater than 0.9 is achieved on a collaborative effort of 4% and 32% from expert and crowd respectively. Further, the cost incurred is less than 10% and 20% (relative cost over an expert only judged pool) when assuming cost models A and B respectively; Expert only judging achieves a similar APCorr measure (0.9) on a 32% judged pool, incurring a cost of 32%. This is a considerable saving of judging resources and time, since a crowdsourcing task is inherently parallel. Using 32% experts and crowd enables a coverage of 64% at a cost just above 45% when assuming model B and considerably lesser on A, however improvement in the rank correlation measure is not observed. Additional 8% and 16% to the expert judgments shows initial gains in cost and rank correlation, but with higher expert contribution, benefit is only observed in coverage at a lower cost. Adding 2% and 4% additional judgments to the did not improve the rank correlation measure, nor did it diminish it, thus is not shown in Figure 3.8.

HCB. The noisier crowd profile as observed on WebTrack 2011 (See Section 3.5.1) performs poorly with crowd only judgments. However, the addition of just 2% expert judgments enables a drastic improvement in rank correlation outperforming the expert only evaluation up to 16%. Increasing expert contribution translates into a consistent improvement for crowd pools of 8% and 16%. However, with an expert contribution of 16% and more, rank correlation does improve relative to the performance of expert only evaluation. While, this may be true, the crowd pools still enable significant coverage at a moderate saving in judging cost. Of note, the collaboration enables a 64% coverage (32% expert + 32% crowd) at a judging cost of only 35% and 40% when assuming cost model A and B respectively. This motivates the use of crowds even though rank correlation does not show an improvement. Crowd contribution levels of 2% and 4% measure similarly on rank correlation as using only experts and similarly skipped in Figure 3.8.

3.6 Conclusion

We present an end to end framework for rapid experimentation of an IR evaluation framework using crowds and expert assessors. We show merit and motivate investigation of machine learning techniques to enable prioritized judging orders which reflect uncertainties in relevance. We present a realistic simulation model that emulates a variety of real world crowds to aid testing prototypes. We propose and validate a collaborative approach that enables a considerable saving in judging effort while still building a scalable and reliable test collection.

Chapter 4

Conclusion

Our work was motivated to develop a synergic IR evaluation framework that accommodated NIST experts and crowd workers. The goals of the framework was to reduce cost, increase speed through parallelization and be scalable while still capable of evaluating IR systems reliably.

To enable reliability from the crowd in Chapter 2 we investigated statistical consensus methods and developed a benchmark in SQUARE to uncover the state-of-the-art in consensus. On the contrary, we found no single method to accommodate each crowd and task type. Surprisingly, DS, a method that was proposed in 1979 performed best on average. The investigation of various crowd datasets enabled developing a realistic crowd simulation model in Chapter 3 to drive rapid experimentation with crowd types in the judging framework.

In Chapter 3 we developed a static ordering of judgements based on expected assessor disagreement which was progressively resilient to judging noise. Our experiments validated the collaborative approach by measuring reliably on rank correlation for both the good and noisy crowd type. Results indicated cost savings when using the 1:10 cost model, while the 1:5 cost model enabled judging coverage and arguably savings in judging time.

By building the evaluation framework with switchable components that is conducive to rapid experimentation across crowd types, we help enable and encourage the community to further experiment with different judging orders, evaluation metrics and aggregation techniques.

Future work will investigate methods that extend our framework to integrate dynamic judging procedures from IR (cf. [3, 12, 45]) with online, adaptive crowdsourcing methods which optimize crowd tasks wrt. an objective metric and a cost budget [19, 31]. In so doing, we can exploit additional recent advances in both IR and human computation fields in order to further our goals of enhancing scalability and reliability of test collection construction using crowds.

Bibliography

[1] O. Alonso and S. Mizzaro. Can we get rid of TREC assessors? Using Mechanical Turk for relevance assessment. In *Proceedings of the SIGIR Workshop on the Future of IR Evaluation*, pages 15–16, 2009.

[2] Nima Asadi, Donald Metzler, Tamer Elsayed, and Jimmy J Lin. Pseudo test collections for learning web search ranking functions. In *SIGIR*, volume 11, pages 1073–1082, 2011.

[3] J.A. Aslam, V. Pavlu, and R. Savell. A unified model for metasearch, pooling, and system evaluation. In *Proceedings of CIKM*, pages 484–491, 2003.

[4] J.A. Aslam, V. Pavlu, and E. Yilmaz. A statistical method for system evaluation using incomplete judgments. In *Proc. of SIGIR*, pages 541–548, 2006.

[5] Javed A. Aslam, Virgiliu Pavlu, and Emine Yilmaz. Measure-based metasearch. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 571–572, New York, NY, USA, 2005. ACM.

[6] Leif Azzopardi, Maarten de Rijke, and Krisztian Balog. Building simulated queries for known-item topics: an analysis using six european languages. In *Proc. SIGIR*, pages 455–462, 2007.

[7] Peter Bailey, Nick Craswell, Ian Soboroff, Paul Thomas, Arjen P. de Vries, and Emine Yilmaz. Relevance assessment: are judges exchangeable and does it matter. In *Proc. SIGIR*, pages 667–674, 2008.

[8] Michael S Bernstein, Greg Little, Robert C Miller, Björn Hartmann, Mark S Ackerman, David R Karger, David Crowell, and Katrina Panovich. Soylent: a word processor with a crowd inside. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 313–322. ACM, 2010.

[9] R. Blanco, H. Halpin, D.M. Herzig, P. Mika, J. Pound, H.S. Thompson, and DT Tran. Repeatable and reliable search system evaluation using crowdsourcing. In *Proceedings of SIGIR*, pages 923–932, 2011.

[10] C. Buckley and E.M. Voorhees. Retrieval evaluation with incomplete information. In *Proceedings of SIGIR*, pages 25–32, 2004.

[11] Chris Buckley, Matthew Lease, and Mark D. Smucker. Overview of the TREC 2010 Relevance Feedback Track (Notebook). In *The Nineteenth Text Retrieval Conference (TREC) Notebook*, 2010.

[12] B. Carterette, J. Allan, and R. Sitaraman. Minimal test collections for retrieval evaluation. In *Proceedings of SIGIR*, pages 268–275, 2006.

[13] Ben Carterette and Ian Soboroff. The effect of assessor error on ir system evaluation. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pages 539–546, New York, NY, USA, 2010. ACM.

[14] Praveen Chandar, William Webber, and Ben Carterette. Document features predicting assessor disagreement. In *Proceedings of the 36th International ACM*

SIGIR Conference on Research and Development in Information Retrieval, SIGIR '13, pages 745–748, New York, NY, USA, 2013. ACM.

[15] C.W. Cleverdon, J. Mills, and M. Keen. *Factors determining the performance of indexing systems*, volume 1. College of Aeronautics, 1966.

[16] P. Clough, M. Sanderson, Jiayu Tang, T. Gollins, and A. Warner. Examining the limits of crowdsourcing for relevance assessment. *Internet Computing, IEEE*, 17(4):32–38, 2013.

[17] Gordon V Cormack, Charles LA Clarke, Christopher R Palmer, and Samuel SL To. Passage-based query refinement:(multitext experiments for trec-6). *Information processing & management*, 36(1):133–153, 2000.

[18] Gordon V Cormack and Aleksander Kolcz. Spam filter evaluation with imprecise ground truth. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 604–611. ACM, 2009.

[19] Peng Dai, Daniel Sabey Weld, et al. Decision-theoretic control of crowd-sourced workflows. In *Proc. AAAI*, pages 1168–1174, 2010.

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

[21] Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proc. WWW*, pages 469–478, 2012.

[22] Catherine Grady and Matthew Lease. Crowdsourcing document relevance assessment with mechanical turk. In *Proceedings of the NAACL HLT 2010 Work-*

shop on Creating Speech and Language Data with Amazon's Mechanical Turk, pages 172–179, 2010.

[23] John Guiver, Stefano Mizzaro, and Stephen Robertson. A few good topics: Experiments in topic set reduction for retrieval evaluation. *ACM Trans. Inf. Syst.*, 27(4), 2009.

[24] Mehdi Hosseini, Ingemar J Cox, Nataša Milić-Frayling, Gabriella Kazai, and Vishwa Vinay. On aggregating labels from multiple crowd workers to infer relevance of documents. In *Advances in Information Retrieval*, pages 182–194. Springer, 2012.

[25] Mehdi Hosseini, Ingemar J Cox, Natasa Milic-Frayling, Trevor Sweeting, and Vishwa Vinay. Prioritizing relevance judgments to improve the construction of ir test collections. In *Proc. CIKM*, pages 641–646, 2011.

[26] Panagiotis G. Ipeirotis. Mechanical Turk: Now with 40.92% spam, 2010. December 16. `http://www.behind-the-enemy-lines.com/2010/12/mechanical-turk-now-with-4092-spam.html`.

[27] P.G. Ipeirotis, F. Provost, and J. Wang. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 64–67. ACM, 2010.

[28] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of KDD*, pages 133–142, 2002.

[29] Hyun Joon Jung and Matthew Lease. Improving Consensus Accuracy via Z-score and Weighted Voting. In *Proceedings of the 3rd Human Computation Workshop (HCOMP) at AAAI*, pages 88–90, 2011.

[30] Hyun Joon Jung and Matthew Lease. Improving Quality of Crowdsourced

Labels via Probabilistic Matrix Factorization. In *Proceedings of the 4th Human Computation Workshop (HCOMP) at AAAI*, 2012.

[31] David R Karger, Sewoong Oh, and Devavrat Shah. Budget-optimal crowdsourcing using low-rank matrix approximations. In *Conference on Communication, Control, and Computing (Allerton)*, pages 284–291. IEEE, 2011.

[32] David R Karger, Sewoong Oh, and Devavrat Shah. Iterative learning for reliable crowdsourcing systems. In *Advances in neural information processing systems*, pages 1953–1961, 2011.

[33] Gabriella Kazai, Nick Craswell, Emine Yilmaz, and Seyed MM Tahaghoghi. An analysis of systematic judging errors in information retrieval. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 105–114. ACM, 2012.

[34] Gabriella Kazai, Jaap Kamps, Marijn Koolen, and Natasa Milic-Frayling. Crowdsourcing for book search evaluation: impact of hit design on comparative system ranking. In *Proc. SIGIR*, pages 205–214, 2011.

[35] Faiza Khan Khattak and Ansaf Salleb-Aouissi. Quality control of crowd labeling through expert evaluation. In *Proceedings of the NIPS 2nd Workshop on Computational Social Science and the Wisdom of Crowds*, 2011.

[36] Beata Beigman Klebanov and Eyal Beigman. Some empirical evidence for annotation noise in a benchmarked dataset. In *Proc. NAACL-HLT*, pages 438–446. Association for Computational Linguistics, 2010.

[37] Michal Kosinski, Yoram Bachrach, Gjergji Kasneci, Jurgen Van-Gael, and Thore Graepel. Crowd iq: measuring the intelligence of crowdsourcing platforms. In *Proceedings of the 3rd Annual ACM Web Science Conference*, pages 151–160. ACM, 2012.

[38] Abhimanu Kumar and Matthew Lease. Modeling annotator accuracies for supervised learning. In *Proceedings of the Workshop on Crowdsourcing for Search and Data Mining (CSDM) at the Fourth ACM International Conference on Web Search and Data Mining (WSDM)*, pages 19–22, 2011.

[39] Balaji Lakshminarayanan and Yee Whye Teh. Inferring ground truth from multi-annotator ordinal data: a probabilistic approach. *arXiv preprint arXiv:1305.0015*, 2013.

[40] Edith Law and L. von Ahn. Human computation. *Synthesis Lectures on AI and Machine Learning*, 5(3):1–121, 2011.

[41] Matthew Lease. On Quality Control and Machine Learning in Crowdsourcing. In *Proceedings of the 3rd Human Computation Workshop (HCOMP) at AAAI*, pages 97–102, 2011.

[42] Michael E Lesk and Gerard Salton. Relevance assessments and retrieval system evaluation. *Information storage and retrieval*, 4(4):343–359, 1968.

[43] Christopher H Lin, Mausam Mausam, and Daniel S Weld. Crowdsourcing control: Moving beyond multiple choice. In *AAAI HCOMP*, 2012.

[44] Chao Liu and Y Wang. Truelabel + confusions: A spectrum of probabilistic models in analyzing multiple ratings. In *Proc. ICML*, 2012.

[45] Alistair Moffat, William Webber, and Justin Zobel. Strategic system comparisons via targeted relevance judgments. In Wessel Kraaij, Arjen P. de Vries, Charles L. A. Clarke, Norbert Fuhr, and Noriko Kando, editors, *SIGIR*, pages 375–382. ACM, 2007.

[46] Barzan Mozafari, Purnamrita Sarkar, Michael J. Franklin, Michael I. Jordan, and Samuel Madden. Active learning for crowd-sourced databases. *CoRR*, abs/1209.3686, 2012.

[47] Quoc Viet Hung Nguyen, Thanh Tam Nguyen, Ngoc Tran Lam, and Karl Aberer. An evaluation of aggregation techniques in crowdsourcing. In *Proceedings of the The 14th International Conference on Web Information System Engineering (WISE 2013)*, 2013.

[48] Jon Noronha, Eric Hysen, Haoqi Zhang, and Krzysztof Z Gajos. Platemate: crowdsourcing nutritional analysis from food photographs. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 1–12. ACM, 2011.

[49] Praveen Paritosh. Human computation must be reproducible. In *CrowdSearch: WWW Workshop on Crowdsourcing Web Search*, pages 20–25, 2012.

[50] Alexander J Quinn and Benjamin B Bederson. Human computation: a survey and taxonomy of a growing field. In *Proc. CHI*, pages 1403–1412, 2011.

[51] Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, and Gerardo Hermosillo Valadez. Learning from crowds. In *Journal of Machine Learning Research 11 (2010) 1297-1322*, MIT Press, 2010.

[52] T. Sakai and N. Kando. On information retrieval metrics designed for evaluation with incomplete relevance assessments. *Information Retrieval*, 11(5):447–470, 2008.

[53] Aashish Sheshadri and Matthew Lease. SQUARE: A Benchmark for Research on Computing Crowd Consensus. In *Proceedings of the 1st AAAI Conference on Human Computation (HCOMP)*, 2013.

[54] Aashish Sheshadri and Matthew Lease. SQUARE: Benchmarking Crowd Consensus at MediaEval. In *Proceedings of the MediaEval 2013 Multimedia Benchmark Workshop*, 2013. CEUR Workshop (cuer-ws.org) Proceedings Vol-1043, ISSN 1613-0073.

[55] Mark D Smucker, James Allan, and Ben Carterette. A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 623–632. ACM, 2007.

[56] P. Smyth, U. Fayyad, M. Burl, P. Perona, and P. Baldi. Inferring ground truth from subjective labelling of venus images. *Advances in neural information processing systems*, pages 1085–1092, 1995.

[57] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proc. EMNLP*, pages 254–263, 2008.

[58] Ian Soboroff. Dynamic test collections: measuring search effectiveness on the live web. In *Proc. SIGIR*, pages 276–283, 2006.

[59] Ian Soboroff, Charles Nicholas, and Patrick Cahan. Ranking retrieval systems without relevance judgments. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, pages 66–73, New York, NY, USA, 2001. ACM.

[60] Eero Sormunen. Liberal relevance criteria of trec-: Counting on negligible documents? In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 324–330. ACM, 2002.

[61] Alexander Sorokin and David Forsyth. Utility data annotation with amazon mechanical turk. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–8. IEEE, 2008.

[62] Wei Tang and Mathew Lease. Semi-Supervised Consensus Labeling for Crowdsourcing. In *Proceedings of the ACM SIGIR Workshop on Crowdsourcing for Information Retrieval*, 2011.

[63] David Tse. The Science of Information: From Communication to DNA Sequencing. In *Talk at the Frontiers of Information Science and Technology (FIST) Meeting*, 2012. December 14. Slides at: `www.eecs.berkeley.edu/~dtse/cuhk_12_v1.pptx`.

[64] Ellen M Voorhees and Donna Harman. Overview of trec 2001. In *Proceedings of the Tenth Text REtrieval Conference (TREC)*, 2001.

[65] E.M. Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. *Information Processing & Management*, 36(5):697–716, 2000.

[66] E.M. Voorhees. Overview of the TREC 2005 robust retrieval track. In *Proceedings of TREC*, 2006.

[67] Jing Wang, Panagiotis Ipeirotis, and Foster Provost. Managing crowdsourcing workers. In *Winter Conference on Business Intelligence*, 2011.

[68] William Webber, Praveen Chandar, and Ben Carterette. Alternative assessor disagreement and retrieval depth. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 125–134. ACM, 2012.

[69] Peter Welinder, Steve Branson, Serge Belongie, and Pietro Perona. The multidimensional wisdom of crowds. In *NIPS*, pages 2424–2432, 2010.

[70] Jacob Whitehill, Paul Ruvolo, Tingfan Wu, Jacob Bergsma, and Javier Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NIPS*, pages 2035–2043, 2009.

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

[72] Emine Yilmaz, Javed A. Aslam, and Stephen Robertson. A new rank correlation coefficient for information retrieval. In *Proc. SIGIR*, SIGIR '08, pages 587–594, New York, NY, USA, 2008.