Exploiting Document Content for Efficient Aggregation of Crowdsourcing Votes

Martin Davtyan
Dept. of Computer Science
ETH Zurich, Switzerland
martin.davtyan@gmail.com

Carsten Eickhoff
Dept. of Computer Science
ETH Zurich, Switzerland
ecarsten@inf.ethz.ch

Thomas Hofmann
Dept. of Computer Science
ETH Zurich, Switzerland
thomas.hofmann@inf.ethz.ch

ABSTRACT

The use of crowdsourcing for document relevance assessment has been found to be a viable alternative to corpus annotation by highly trained experts. The question of quality control is a recurring challenge that is often addressed by aggregating multiple individual assessments of the same topic-document pair from independent workers. In the past, such aggregation schemes have been weighted or filtered by estimates of worker reliability based on a multitude of behavioural features. In this paper, we propose an alternative approach by relying on document information. Inspired by the clustering hypothesis of information retrieval, we assume textually similar documents to show similar degrees of relevance towards a given topic. Following up on this intuition, we propagate crowd-generated relevance judgments to similar documents, effectively smoothing the distribution of relevance labels across the similarity space.

Our experiments are based on TREC Crowdsourcing Track data and show that even simple aggregation methods utilizing document similarity information significantly improve over majority voting in terms of accuracy as well as cost efficiency.

Categories and Subject Descriptors

H.5 [**Information Retrieval**]: Evaluation of retrieval results—*Relevance assessment*;
H.4 [**World Wide Web**]: Web applications—*Crowdsourcing*

General Terms

Theory, Experimentation

Keywords

Crowdsourcing, Clustering Hypothesis, Relevance Assessment

CIKM'15, October 19 - 23, 2015, Melbourne, VIC, Australia.

DOI: http://dx.doi.org/10.1145/2806416.2806460.

1. INTRODUCTION

Evaluation of retrieval performance is a crucial step in the overall IR system design process. In order to ensure frequent repeatability of tests, many researchers and practitioners rely on static test collections with known relevance judgments for pairs of topics and documents. The creation of such resources, especially at large scale, can require considerable amounts of time and money as an expensive group of trained domain experts carefully judges the relevance of individual pairs [29].

Crowdsourcing is defined as the practice of obtaining content from a large (typically, online) community, rather than from traditional employees. There are many ways in which individuals can be incentivized to offer their work force on crowd markets. While altruistic motives [3], community credibility [33], and entertainment [11] form valid motivating factors, paid crowd labour tends to be one of the most broadly applicable schemes. Crowdsourcing platforms such as CrowdFlower or Amazon's Mechanical Turk take the role of intermediaries between *requesters* and *workers*. Assignments are typically given out and paid for in the form of small, atomic units, so-called *Human Intelligence Tasks* (HITs). In the case of our IR evaluation scenario, making a single binary relevance assessment between a topic and a document can be considered a HIT.

Quality control is a traditional challenge that crowdsourcing requesters have to address when drawing from the considerable power of the crowd. While most workers attempt to truthfully complete tasks, there are frequent reports of workers showing sloppy or random judging behaviour in order to increase their time efficiency [10]. A widely accepted way of overcoming this obstacle is to present the same HIT to multiple workers and subsequently aggregate their submissions. To continue with our IR test collection creation example, each query-document pair is presented to multiple workers and its final relevance label is determined by means of an aggregation method such as interpolation or majority voting. For many tasks, including document relevance assessment, this practice has effects that go beyond mere filtering of spam submissions but can also account for subjective differences in judgments across workers.

Instead of uniformly merging raw votes, much work has been dedicated into estimating worker reliability based on their past accuracy, judging behaviour, or topic affinity. Subsequently, this form of worker information can be used to bias the aggregation process towards the most reliable workers or to empower active learning schemes in which the most suitable worker for each task is to be selected. There is, how-

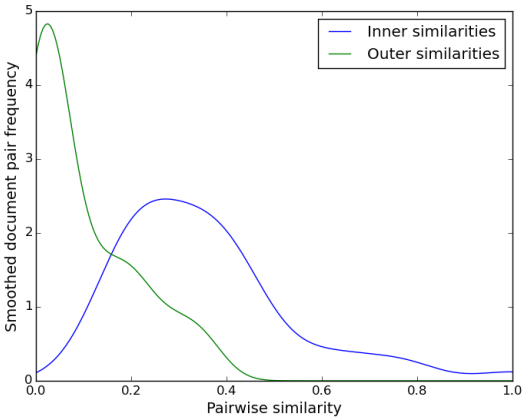

Figure 1: Gaussian kernel density estimate of the document similarity distribution for topic 20542

ever, another, largely untapped source of information, the document's content. For example, one could exploit similarities between documents to aggregate worker votes in an efficient manner.

Consider a cosine similarity between the tf–idf representations of documents. Figure 1 shows the distribution of similarities a) between relevant documents (inner similarities) and b) between relevant and irrelevant documents (outer similarities) for topic 20542 of the TREC Crowdsourcing Track 2011. We can clearly observe how relevant documents share much stronger commonalities than irrelevant ones. In this paper, we exploit this well-known *Clustering Hypothesis* by propagating relevance assessments to "nearby" neighbours for the purpose of vote aggregation.

The novel contributions of this paper are threefold: 1) We present a systematic overview of the spread of crowdsourced relevance labels across a textual similarity space, supporting the validity of the clustering hypothesis for label aggregation. 2) We introduce three content-aware vote aggregation methods. Beginning with light and moderate modifications to the standard majority voting approach, we finally describe the use of Gaussian process classification models for this purpose. 3) In a set of experiments based on historic submissions to the TREC 2011 Crowdsourcing Track, we demonstrate the merit of our methods in terms of cost-efficiency and accuracy in comparison to content agnostic methods.

The remainder of this paper is structured as follows: Section 2 gives an overview of related work dedicated to crowdsourced relevance assessments and quality control mechanisms. Section 3 formally introduces our methods and discusses relevant technical considerations. Our experimental setup and results are described in Section 4, before Section 5 concludes our investigation with a brief discussion of practical implications as well as future directions inspired by our findings.

2. RELATED WORK

Ever since the introduction of the Cranfield experiments [28], test collections have been one of the pillars of IR system evaluation. Traditionally, such collections are created by trained professionals in controlled lab environments. In the IR community, the Text REtrieval Conference (TREC) [29] supports one of the most widely known efforts to creating such collections. In the face of constantly growing demands in terms of test collection diversity and scale, there have been numerous attempts at reducing the considerable cost involved in corpus creation and annotation. Most notably, previous work proposes designing more robust performance measures [4], selecting the right subset of documents for evaluation [6] and inferring implicit judgments from user interaction logs [16].

With the rise of crowdsourcing, there has been an extensive line of research dedicated to using this new channel for the creation and annotation of IR test collections. An early set of experiments [21, 1, 20, 12] find that aggregated labels of multiple untrained crowd workers can reach a quality comparable to that of a single highly trained NIST assessor.

While this alternative labour market has been shown to be time and cost efficient [11], researchers have less control over the way in which relevance judgments are created. Traditionally, inaccurate judgments as well as spam submissions have been a major challenge in the crowdsourcing process. There are many effective quality control schemes, ranging from aggregating the results of independent workers to the use of honey pot questions [13, 15, 10]. Marshall et al. [24] highlight the importance of engaging HIT design on result quality. Active learning techniques have been demonstrated to greatly improve the worker-task allocation step [32].

Significant effort has been made into estimating worker reliability based on various behavioural and demographic traits. Tang and Lease [26] introduce a semi-supervised method for reliability estimation based both on labeled as well as unlabeled examples. Kazai et al. [18, 19] group workers into 5 different classes and study their respective judgment reliability and behaviour. Based on social media profiles, Difallah et al. [9] model worker topic affinities, enabling them to assign tasks to workers with matching interest, resulting in significantly improved result quality. Karger et al. [17] propose a joint model for iteratively learning worker reliability and aggregating votes by means of approximate belief propagation. Blanco et al. [2] investigate the robustness of crowdsourced relevance assessments over time, finding that repeated labeling efforts produce stable results even as longer periods of time elapse.

Another prominent source of evidence can be found in the analysis of systematic judgment behaviour. Following Dawid and Skene [8], who investigated disagreements between diagnoses posed by multiple individual medical doctors, there have been several successful attempts at harnessing similar methods for crowdsourcing quality assurance [30, 14]. In this way, reliable workers that make occasional mistakes can be accurately separated from spammers that select answers at random or follow other, more sophisticated, cheating strategies.

Several scientific workshops have been dedicated to pursuing how to use crowdsourcing effectively and efficiently [7, 23]. Most notably, TREC 2011 for the first time offered a dedicated crowdsourcing track [22], addressing the crowdsourced collection of document relevance assessments.

While previous approaches have been successfully using worker demographics and behaviour in order to predict judgment accuracy, in this paper, we propose to investigate the content of the documents being judged. Inspired by the clus-

tering hypothesis of information retrieval [27], we assume similar documents to show similar degrees of relevance towards a given topic. Our experiments show significantly improved label aggregation performance at lower overall cost, when relying on this source of evidence.

3. METHODOLOGY

In this section, we formally introduce the problem statement alongside our proposed method. For each topic, there is a set documents D, that each require relevance labels. For every document $d_i \in D$, we can, via a crowdsourcing platform, request binary relevance judgments (votes) v_{ij}. Individual votes are encoded such, that $v_{ij} = 1$ denotes "relevance" and $v_{ij} = 0$ denotes "non-relevance" of the given topic-document pair. To account for errors or worker subjectivity in the creation of individual votes, a final processing step specifies vote aggregation methods to produce overall relevance labels $r_i \in \{0, 1\}$ on the basis of the the collected raw votes. For practical reasons, suppose that we can query the crowd for a worker vote on any document d_i at any given time. Crowdsourcing can then be represented as a process of iterative requesting of votes, where relevance labels can be aggregated at every step:

Algorithm 1 CROWDSOURCED RELEVANCE ASSESSMENT

```
for k ← 1 ... K do
    d_i ← PickDocument(ω)
    V_i ← V_i ∪ RequestVote(d_i)
    R ← AggregateVotes(ω, D)
end for
```

Where K delimits the number of iterations, taking the role of a budget parameter. V_i denotes the set of all raw votes v_{ij} requested for document d_i, ω is the super set of all currently requested votes across D and R is the set of all final relevance labels. There are three fundamental components to the generalized crowdsourcing-based relevance assessment process described in Algorithm 1. PICKDOCUMENT selects the next document to request a vote for. To this end, we randomly sample among those documents that currently received the lowest overall number of votes ($d_i = D(\arg\min(V_i))$), effectively introducing a weighted round robin scheduling. REQUESTVOTE details the exact procedure under which a new vote is requested (e.g., crowdsourcing platform, interface, gold standard, etc.). In this work, we rely on a large set of existing votes collected for the TREC 2011 Crowdsourcing Track [22] from which we randomly sample with replacement in order to obtain an infinite supply of votes.

Since the focus of our work resides on content-aware vote aggregation methods, we do not alter the remaining components in the course of our experiments. Please refer to Section 5 for a discussion of future work on content-aware active learning schemes that may indeed want to introduce alternative realizations of PICKDOCUMENT and REQUESTVOTE.

3.1 Tie Breaking

As stated earlier, we require aggregation methods to produce relevance labels for *all* documents at *every* iteration. As a consequence, there can be ties when:

- Some documents have not been judged by workers, leaving us without explicit evidence of relevance.
- The aggregation method received both positive and negative votes on document relevance, and is unable to decide. In case of standard majority voting, this can occur simply when the numbers of "relevant" and "non-relevant" votes for a document are equal.

These cases can be handled similarly for all proposed aggregation methods. In this work, we resolve ties by *selecting between "relevance" and "non-relevance" by tossing a fair coin.* Ideally, we would want to toss a biased coin informed by the underlying probability of relevance in the collection, but for the sake of realism, we assume the aggregation method to only have access to the subset of votes currently yielded by the crowdsourcing process. Since this assumption affects all proposed methods and baselines equally, we do not expect it to introduce any systematic bias towards favoring either of the methods. In fact, as we will see in Section 4, our experimental collection shows a near-uniform distribution of relevant and irrelevant documents.

3.2 Baseline: Majority Voting

As an intuitive performance baseline, we rely on the *de facto* standard in crowdsourcing vote aggregation [10], majority voting. Let us consider the set of votes V_i received for document d_i as a realization of a sequence of Bernoulli trials. If we denote the amount of "relevant" votes received for the topic-document pair as N_i,

$$N_i \sim Binomial(|V_i|, p_i)$$

Therefore p_i can be estimated as:

$$\widehat{p_i} = \frac{N_i}{|V_i|} = \overline{V_i}$$

Here we use a vertical bar to denote the arithmetic mean across all votes in the set. We can now present a Majority Vote aggregation function in an algorithmic form, see Algorithm 2.

Algorithm 2 MAJORITYVOTE

```
for all i ∈ I do
    if |V_i| = 0 then
        p̂_i ← 0.5
    else
        p̂_i ← V̄_i
    end if
end for
```

3.3 Document similarity

As stated earlier, the novel methods introduced in this work are based on the notion of similarity between textual documents. To this end, we represent each document d_i by its tf–idf vector $T(d_i)$ and define pairwise similarity between two documents d_a and d_b in terms of *cosine similarity* ρ between their vector representations:

$$\rho(d_a, d_b) = \frac{T(d_a) \cdot T(d_b)}{|T(d_a)||T(d_b)|},$$

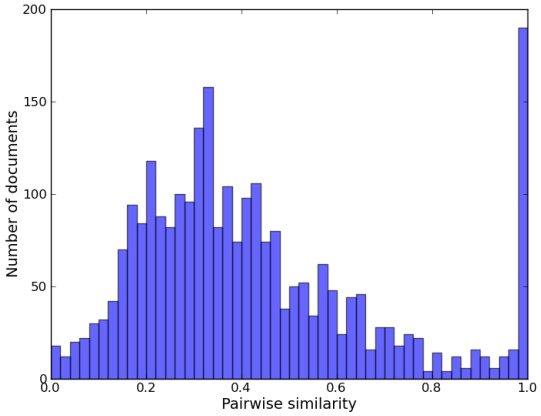

Figure 2: (Inner) Similarity between relevant documents averaged across queries.

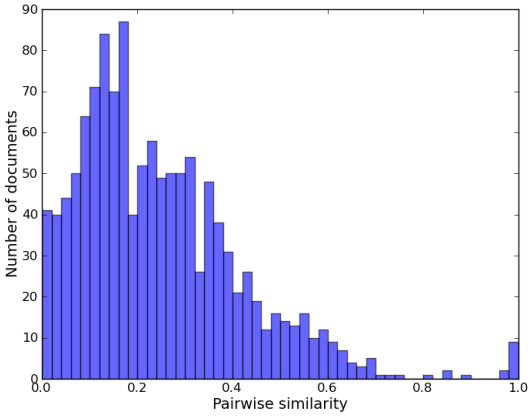

Figure 3: (Outer) Similarity between relevant and irrelevant documents averaged across queries.

where $|\cdot|$ denotes the Euclidean norm of a vector. Documents are processed as they are, without any form of vocabulary pruning, stop word removal, stemming etc. To validate our choice of distance metric for the task of separating relevant documents from irrelevant ones, let us first test the clustering hypothesis. We apply the method which was originally proposed by van Rijsbergen and Spärck Jones [27] for identifying document collections for which retrieval could yield meaningful results. We consider a set $D_r \subset D$ of documents which are relevant to a particular topic and compute two sets of pairwise similarities:

1. "Inner" similarities $S_I = \{\rho(d_u, d_v) | d_u, d_v \in D_r\}$ - similarities between relevant documents.
2. "Outer" similarities $S_O = \{\rho(d_u, d_v) | d_u \in D_r, d_v \notin D_r\}$ - similarities between relevant documents and irrelevant ones.

Our investigation is based on the ClueWeb09-T11Crowd collection, a subset of the full ClueWeb09 dataset [5], and uses NIST-created TREC 2011 Crowdsourcing Track topics and relevance judgments. We join these respective sets across all queries in the document collection and produce two histograms, where inner and outer similarities are plotted against their relative frequency in corresponding joined sets. The results are shown in Figures 2 and 3.

The separation between the two histograms supports the clustering hypothesis and justifies our choice of document similarity metric. In Figure 2 one can see the spike indicating there is a relatively high number of document pairs with similarities close to 1. Manual inspection of some randomly selected pairs hints that these could be Wikipedia pages redirecting to one common page, which differ only in the "Redirected from" header. The presence of such near duplicates in the dataset may pose problems in performance estimation. The aggregation methods we propose attempt to use relevance judgments from similar documents when aggregating final estimates. For pairs of documents which are very similar to each other this strategy is likely to be very successful. Hence, the fact that the dataset contains a significant amount of document pairs that are near duplicates means that the accuracy gain may be *overestimated*.

To investigate the effect of such document pairs with artificially high similarity scores, we manually inspect their distribution across the data. Queries "20694" and "20584" show the highest number of pairs close to a similarity of 1. One could argue that such topics are outliers and should be disregarded. However, even after removing these topics, there is no noticeable change to the spiking behaviour or the overall pattern. Hence, we do not consider these topics outliers, rather, having a number of near-duplicate documents appears to be a specific population property of a sizeable Web document collection.

3.4 Aggregation Methods

In the following, we will introduce three content-aware aggregation methods that serve as alternative realizations of the AggregateVotes component in Algorithm 1.

Majority Voting with Nearest Neighbor

Our first candidate method, Majority Voting with Nearest Neighbor (MVNN), is a straightforward extension of the baseline method that draws evidence from the single closest neighbor in tf–idf space. Consider a permutation $O^i = \{o^i(1) \dots o^i(|I|)\}$ which sorts all documents by decreasing similarity to document d_i:

$$a, b \in I : a < b \rightarrow \rho(d_{o^i(b)}, d_i) \leq \rho(d_{o^i(a)}, d_i)$$

While $o^i(1) = i$ refers to the document itself, $o^i(2)$ is the index of the *closest neighbor* of document d_i. We then define a similarity threshold parameter $\rho_s \in [0, 1]$ to control for topical drift. Algorithm 3 details the aggregation algorithm that for every document d_i merges votes V_i with all votes $V_{o^i(2)}$ requested for its single nearest neighbor $d_{o^i(2)}$ if that neighbor's similarity to d_i is greater than ρ_s. In this way, we locally smooth relevance labels across the tf–idf space.

Merging Enough Votes

As an extension to the previous method, we would like to use evidence from multiple neighbors instead of just one. For example, such an extension could define the number of neighbors to consider as an additional parameter. Additionally, we may even want to vary the amount of neighbors

Algorithm 3 MajorityVoteWithNearestNeighbor

Parameters: similarity threshold ρ_s
for all $i \in I$ **do**
  if $|V_i| = 0$ **then**
    $\widehat{p}_i \leftarrow 0.5$
  else
    if $\rho(d_i, d_{o^i(2)}) > \rho_s$ **then**
      $\widehat{p}_i \leftarrow \overline{V_i \cup V_{o^i(2)}}$
    else
      $\widehat{p}_i \leftarrow \overline{V_i}$
    end if
  end if
end for

employed in the aggregation for a particular document depending on the number of votes already requested. For documents which did not yet receive many votes, we rely more heavily on votes in the neighborhood. We formalize this intuition by requiring a desired overall amount of votes per document rather than explicitly fixing the number of neighbors as a parameter. If the document has less votes than required, we merge votes with the closest neighbors until the required vote count is reached. This Merge Enough Votes (MEV) strategy is preferable since it ensures comparable amounts of information to be used in the label aggregation of each document, even if this requires relying on a wider neighborhood. See Algorithm 4:

Algorithm 4 MergeEnoughVotes

Parameters: votes per document required C
for all $i \in I$ **do**
  $U_i \leftarrow V_i$
  if $|U_i| < C$ **then**
    for $k = 2 \ldots |I|$ **do**
      $U_i \leftarrow U_i \cup V_{o^i(k)}$
      if $|U_i| \geq C$ **then**
        break
      end if
    end for
  end if
  $\widehat{p}_i \leftarrow \overline{U_i}$
end for

Gaussian Process Aggregation

In this work, we aim to improve the aggregation of crowdsourced relevance votes by utilizing the similarity between documents in content space. The previously proposed heuristic methods (MVNN and MEV) achieve this by incorporating votes from a number of highly similar neighboring documents.

Gaussian process classification [31] is a well-known discriminative method, in which labels are inferred by modeling "similarity" of points in a feature space. We use this method to utilize votes available for *all* documents, as opposed to just immediate neighbors. To consider the GP classification method a formal extension of the proposed heuristics, we rely on the same notion of similarity, Cosine similarity between the tf–idf vectors becomes our *linear* covariance function in tf–idf space.

We specify the Gaussian process with a constant mean function and a linear covariance function:

$$m(x) = c,$$

$$k(x, x') = x \cdot x',$$

where $\cdot$ denotes a scalar product. We train a GP classifier on a set P of all available (document, vote) pairs, where the tf–idf representation of a document $T(d_i)$ becomes our feature vector, with the raw binary vote $v \in \{0, 1\}$ as the class label. We then retrieve posterior probabilities of relevance $\widehat{p}_i$ for all documents in a topic. Algorithm 5 illustrates this process. The exact inference for the posterior Gaussian Process is not feasible, hence an approximate solution is found using an expectation propagation algorithm [25] with cumulative Gaussian likelihood function. The only calculated hyperparameter is the constant mean c.

Algorithm 5 GaussianProcessAggregation

$P \leftarrow \varnothing$
for all $i \in I$ **do**
  for all $v \in V_i$ **do**
    $P \leftarrow P \cup (T(d_i), v)$
  end for
end for
GPClassifier.train(P)
for all $i \in I$ **do**
  $\widehat{p}_i \leftarrow$ GPClassifier.posterior($T(d_i)$)
end for

4. EXPERIMENTS

In this section, we describe our experimental collection alongside the evaluation strategy and compare the performance of the individual methods. We show that content-aware aggregation methods attain the goal of outperforming content-agnostic approaches such as majority voting.

4.1 Data

The TREC Crowdsourcing Track [22], hosted between 2011 and 2013, is dedicated to investigating the use of crowdsourcing for search engine evaluation. In 2011, participating teams were offered two tasks:

1. *Assessment*, in which participants are to gather individual relevance judgments through a crowdsourcing platform.
2. *Consensus*, in which teams have to perform *label aggregation* over a set of individual worker judgments to produce a final overall relevance label.

The consensus task is highly relevant to the vote aggregation problem addressed here. In 2012 and 2013 tracks, there was no explicit separation between assessment and aggregation. Therefore, in this paper, we adopt the data and evaluation procedure from the Crowdsourcing Track 2011 and focus on the problem setting of the consensus task. The ClueWeb09-T11Crowd collection is a subset of the full ClueWeb09 dataset [5]. Every document in the collection is a uniquely identified Web page represented by the page

URL, the full page text, comparable to what a user would see when opening the page in a browser, the HTML code of the page alongside the HTML header. Out of this information, we only use the full page text. Although structural information contained in the HTML code could be potentially useful in the future, that sort of application would require a different choice of similarity metric.

We use the original *relevance judgments* submitted by TREC 2011 participants. They were given for 30 different topics, such as "free email directory" or "growing tomatoes". For every topic, there is a set of approximately 100 documents. For every document there are on average 15 relevance judgments from individual workers, although some topic-document pairs have fewer affiliated judgments. For two topics (20644 and 20922) there were documents with as few as one single vote. Since such singleton "pools" of votes are very brittle and make for a poor representation of human knowledge, we exclude these two outlier topics from our investigation, leaving us with 28 functional ones.

In the 2011 Crowdsourcing Track evaluation was based on two benchmark annotations, expert judgments from NIST assessors as well as aggregated consensus labels gathered across all participating teams. In this work, we solely rely on data from the NIST assessors as ground truth. It consists of 395 relevance judgments and most topics contain ten to twenty documents with such ground truth labels. After the track participants submitted the aggregated judgments they were evaluated using the benchmark sets. For every submission precision, recall, accuracy and specificity were measured. For the available ground truth labels, both relevance classes have similar orders of magnitude – 68% of 395 labels are "relevant". While the "relevant" class is larger across all topics, there is only one case in which it represents as much as 80% of labels. We therefore concentrate on accuracy as a performance measure, since, in our case, it is expressive of the classifier's performance.

The average differences between the ground truth judgment and the majority vote estimate for a document vary from topic to topic. Averaged across topics the difference is 0.15 (on a scale from -1 to 1). Intuitively, this indicates that the worker judgments aggregated with majority voting are a reasonable estimate of the true relevance and indeed provide a meaningful baseline for our methods. Nevertheless, for four topics the average difference exceeds 0.30, for one of them (20922) reaching the value of 0.55. For these topics it appears less likely to attain a reasonable accuracy with any aggregation method due to the low overall agreement between crowd and experts.

4.2 Results

Our experimental setup is based on the general iterative crowdsourcing procedure presented in Algorithm 1. At each step, one of the documents with the least amount of votes is sampled at random and a vote is requested. Since we work with a stale collection of votes, we sample with replacement from the pool of raw votes submitted for document d_i to prevent running out of votes in case of sparsely-annotated topics. We measure accuracy of aggregated relevance estimates at every step. Optimal settings of MVNN ($C = 0.5$) and MEV($\rho_s = 1.0$) parameters were determined empirically in a dedicated set of experiments while Gaussian process hyperparameter c is determined within each learning iteration. Figure 4 displays aggregation performance as a function of the number of votes per document within a given topic. Each plotted performance measurement represents the mean accuracy across 50 sampling randomizations of the crowdsourcing process. Within each randomized run, the individual methods are evaluated on the exact same realizations of the various sampling processes in order to ensure comparability of our findings.

There are several interesting trends to be observed. MVNN closely follows the performance curve of the majority voting baseline. While less than 2 votes per document are requested, the neighborhood information introduces a slight performance gain that levels out and is eventually reversed as more document-specific votes are procured. GP and MEV start at a significant performance offset, yielding significant improvements when only few votes are requested. With every additional requested vote,we can note the relative advantage of both methods shrinking. While MEV maintains a narrow lead, GP performance eventually dwindles. At approximately three votes per document, all methods reach a stable accuracy level of approximately 0.75 which is not significantly improved by further votes. This finding is in line with previous observations, made, e.g., by Vuurens et al. [30], who find that requesting more than three votes on average does not improve labelling performance. We believe that the observed method ranking stems from the particular scope at which neighborhood information is considered by the individual methods. MV shows the steepest performance increase, drawing new information from every new vote. MVNN has an initial advantage by using the single closest neighbor for smoothing. As sufficiently many local votes are available, this smoothing effect, however, begins to turn into noise. Gaussian processes use the widest neighborhood range, drawing from *all* available labels which results in a very strong early performance, but serves for noisy labels later on. MEV offers a good trade-off between strong early performance and good noise robustness since it effectively defaults back to local majority voting as soon as more local votes are available.

In this work, we focus especially on the extremely low-budget scenario in which aggregation effectiveness is subjected to tight cost-efficiency bounds. Settings like these are frequently encountered when large-scale problems are studied. To get a better impression of the relative comparison between the four methods as votes are scarce, Table 1 investigates aggregation performance per topic at a cut-off of 1 vote per document. It should be noted that in this particular case, majority voting (MV) defaults to the case where the single requested crowd label is believed to be true for every document. Statistically significant improvements over the content-agnostic baseline MV are denoted by an asterisk character. Methods that outperform *all* competitors at significance level are indicated by a hash symbol. Statistical significance was tested using a Wilcoxon signed rank at $\alpha < 0.05$-level. Again, all presented results are mean values across 50 randomizations.

We can note that in the extremely resource-constrained setting, the wide-coverage neighborhood model used by GP performs best. Out of 28 topics, the method yields the highest overall accuracy in 18 cases, often significantly outperforming all competing methods. MEV is the runner-up with 8 overall best accuracies. MVNN typically introduces only mild improvements over the majority voting baseline, few of which turn out to be statistically significant.

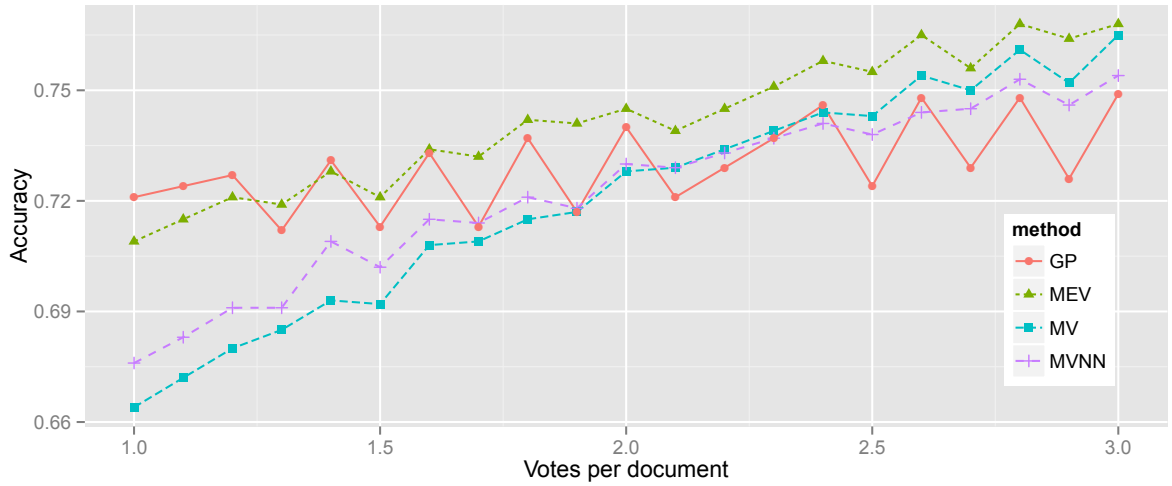

Figure 4: Aggregation performance across topics

Table 1: Performance comparison in terms of accuracy at 1 vote per document for each topic.

Topic	MV	MVNN(0.5)	MEV(1)	GP
20424	0.594	0.608	0.602	**0.702**$^{\#}$
20488	0.667	0.722	0.744*	**0.792**$^{\#}$
20542	0.676	0.673	0.707	**0.720***
20584	0.689	0.761*	0.793*	**0.856**$^{\#}$
20636	0.716	0.774*	**0.784**$^{\#}$	0.712
20642	0.693	0.756*	0.789*	**0.877**$^{\#}$
20686	**0.693**	0.639	0.665	0.688
20690	0.703	0.744*	0.766*	**0.830**$^{\#}$
20694	0.726	0.746	0.783*	**0.801***
20696	0.562	0.622*	0.634*	**0.734**$^{\#}$
20704	0.670	0.790*	0.804*	**0.942**$^{\#}$
20714	0.808	0.818	0.868*	**0.884***
20764	**0.676**	0.600	**0.676**	0.448
20766	0.796	0.794	**0.852***	0.846*
20778	0.652	0.662	0.654	**0.694**
20780	0.641	0.732*	0.761*	**0.771***
20812	0.688	0.695	**0.740***	0.727*
20814	0.792	0.770	0.858*	**0.906**$^{\#}$
20832	0.630	0.642	**0.658**	0.656
20910	0.661	0.643	**0.707**$^{\#}$	0.681
20916	0.650	0.638	0.628	**0.706**$^{\#}$
20932	0.576	0.540	0.577	**0.696**$^{\#}$
20956	0.616	0.646	0.666*	**0.824**$^{\#}$
20958	0.610	0.556	0.636	**0.638**
20962	0.552	0.610*	0.596*	**0.622***
20972	0.668	0.632	**0.692**	0.492
20976	0.632	0.620	**0.676**$^{\#}$	0.616
20996	**0.542**	0.504	0.534	0.326
Overall	0.664	0.676	0.709*	**0.721***

5. DISCUSSION & CONCLUSION

In this paper, we demonstrated the use of document similarity information for aggregating crowdsourced relevance assessments. Following the intuition that textually similar documents should show similar degrees of relevance towards a given query, we propagate crowdsourced relevance judgements across documents in order to infer the relevance of those documents that have not yet received (enough) explicit votes. In a series of experiments based on the data and guidelines of the TREC 2011 Crowdsourcing Track, we show that even straight-forward methods informed by document similarity estimates significantly outperform commonly used majority voting schemes in terms of both label accuracy as well as cost efficiency.

We investigated three novel aggregation schemes relying on varying scopes of neighborhood information. Gaussian process classification considers the full available set of all votes, resulting in competitive performance in cold-start scenarios with very few available votes. The two heuristics MVNN and MEV are more conservative in their use of neighborhood information, relying on fewer, highly similar neighbors, making them suitable for resource-rich scenarios in which many raw votes are available for every document.

A particular caveat when using methods like these lies in the danger of biasing the created labels too strongly towards the same intuition underlying common retrieval systems (i.e., tf–idf locality of relevant documents). While this is a valid concern, we believe that it can easily be controlled for by making only careful use of local neighborhood information instead of long-distance propagation of relevance labels.

There are several exciting directions for future work. In this paper, we rely solely on document information, ignoring evidence of worker reliability. In the future it would be interesting to investigate to which degree these orthogonal sources of information can be joined to increase overall performance. Currently, our method does not include any active selection of documents to request votes for next. It would be interesting to holistically model all documents to be evaluated for a given topic and carefully select which

concrete documents to require judgements for in an active learning scheme such that the entire system is benefitted most.

6. REFERENCES

[1] Omar Alonso, Daniel E Rose, and Benjamin Stewart. Crowdsourcing for relevance evaluation. In *ACM SigIR Forum*, volume 42, pages 9–15. ACM, 2008.

[2] Roi Blanco, Harry Halpin, Daniel M Herzig, Peter Mika, Jeffrey Pound, Henry S Thompson, and Thanh Tran Duc. Repeatable and reliable search system evaluation using crowdsourcing. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 923–932. ACM, 2011.

[3] Daren C Brabham. Moving the crowd at threadless: Motivations for participation in a crowdsourcing application. *Information, Communication & Society*, 13(8):1122–1145, 2010.

[4] Chris Buckley and Ellen M Voorhees. Retrieval evaluation with incomplete information. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 25–32. ACM, 2004.

[5] Jamie Callan, Mark Hoy, Changkuk Yoo, and Le Zhao. Clueweb09 data set, 2009.

[6] Ben Carterette, James Allan, and Ramesh Sitaraman. Minimal test collections for retrieval evaluation. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 268–275. ACM, 2006.

[7] Vitor R Carvalho, Matthew Lease, and Emine Yilmaz. Crowdsourcing for search evaluation. In *ACM Sigir forum*, volume 44, pages 17–22. ACM, 2011.

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

[9] Djellel Eddine Difallah, Gianluca Demartini, and Philippe Cudré-Mauroux. Pick-a-crowd: tell me what you like, and i'll tell you what to do. In *Proceedings of the 22nd international conference on World Wide Web*, pages 367–374. International World Wide Web Conferences Steering Committee, 2013.

[10] Carsten Eickhoff and Arjen P de Vries. Increasing cheat robustness of crowdsourcing tasks. *Information retrieval*, 16(2):121–137, 2013.

[11] Carsten Eickhoff, Christopher G Harris, Arjen P de Vries, and Padmini Srinivasan. Quality through flow and immersion: gamifying crowdsourced relevance assessments. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 871–880. ACM, 2012.

[12] Catherine Grady and Matthew Lease. Crowdsourcing document relevance assessment with mechanical turk. In *Proceedings of the NAACL HLT 2010 workshop on creating speech and language data with Amazon's mechanical turk*, pages 172–179. Association for Computational Linguistics, 2010.

[13] Matthias Hirth, Tobias Hoßfeld, and Phuoc Tran-Gia. Cheat-detection mechanisms for crowdsourcing. *University of Würzburg, Tech. Rep*, 4, 2010.

[14] 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.

[15] Panos Ipeirotis. Crowdsourcing using mechanical turk: quality management and scalability. In *Proceedings of the 8th International Workshop on Information Integration on the Web: in conjunction with WWW 2011*, page 1. ACM, 2011.

[16] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM, 2002.

[17] David R Karger, Sewoong Oh, and Devavrat Shah. Budget-optimal task allocation for reliable crowdsourcing systems. *Operations Research*, 62(1):1–24, 2014.

[18] Gabriella Kazai, Jaap Kamps, and Natasa Milic-Frayling. Worker types and personality traits in crowdsourcing relevance labels. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1941–1944. ACM, 2011.

[19] Gabriella Kazai, Jaap Kamps, and Natasa Milic-Frayling. An analysis of human factors and label accuracy in crowdsourcing relevance judgments. *Information retrieval*, 16(2):138–178, 2013.

[20] Gabriella Kazai and Natasa Milic-Frayling. On the evaluation of the quality of relevance assessments collected through crowdsourcing. In *SIGIR 2009 Workshop on the Future of IR Evaluation*, page 21, 2009.

[21] Aniket Kittur, Ed H Chi, and Bongwon Suh. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 453–456. ACM, 2008.

[22] Matthew Lease and Gabriella Kazai. Overview of the trec 2011 crowdsourcing track. In *Proceedings of the text retrieval conference (TREC)*, 2011.

[23] Matthew Lease and Emine Yilmaz. Crowdsourcing for information retrieval. In *ACM SIGIR Forum*, volume 45, pages 66–75. ACM, 2012.

[24] Catherine C Marshall and Frank M Shipman. The ownership and reuse of visual media. In *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*, pages 157–166. ACM, 2011.

[25] Thomas P. Minka. Expectation propagation for approximate bayesian inference. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, UAI'01, pages 362–369, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[26] Wei Tang and Matthew Lease. Semi-supervised consensus labeling for crowdsourcing. In *SIGIR 2011 workshop on crowdsourcing for information retrieval (CIR)*, 2011.

[27] Cornelis J van Rijsbergen and Karen Spärck Jones. A test for the separation of relevant and non-relevant documents in experimental retrieval collections. *Journal of Documentation*, 29(3):251–257, 1973.

[28] Ellen M Voorhees. The philosophy of information retrieval evaluation. In *Evaluation of cross-language information retrieval systems*, pages 355–370. Springer, 2002.

[29] Ellen M Voorhees, Donna K Harman, et al. *TREC: Experiment and evaluation in information retrieval*, volume 1. MIT press Cambridge, 2005.

[30] Jeroen Vuurens, Arjen P de Vries, and Carsten Eickhoff. How much spam can you take? an analysis of crowdsourcing results to increase accuracy. In *Proc. ACM SIGIR Workshop on Crowdsourcing for Information Retrieval (CIR'11)*, pages 21–26, 2011.

[31] Christopher KI Williams and Carl Edward Rasmussen. Gaussian processes for machine learning. *the MIT Press*, 2(3):4, 2006.

[32] Yan Yan, Glenn M Fung, Rómer Rosales, and Jennifer G Dy. Active learning from crowds. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 1161–1168, 2011.

[33] Yu Zhang and Mihaela van der Schaar. Reputation-based incentive protocols in crowdsourcing applications. In *INFOCOM, 2012 Proceedings IEEE*, pages 2140–2148. IEEE, 2012.