Collecting High Quality Overlapping Labels at Low Cost

Hui Yang[*]
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA, 15213
huiyang@cs.cmu.edu

Anton Mityagin
Microsoft Bing
One Microsoft Way
Redmond, WA 98052
mityagin@gmail.com

Krysta M. Svore
Microsoft Research
One Microsoft Way
Redmond, WA 98052
ksvore@microsoft.com

Sergey Markov
Microsoft Bing
One Microsoft Way
Redmond, WA 98052
sergey.markov@microsoft.com

ABSTRACT

This paper studies quality of human labels used to train search engines' rankers. Our specific focus is performance improvements obtained by using overlapping relevance labels, which is by collecting multiple human judgments for each training sample. The paper explores whether, when, and for which samples one should obtain overlapping training labels, as well as how many labels per sample are needed. The proposed selective labeling scheme collects additional labels only for a subset of training samples, specifically for those that are labeled relevant by a judge. Our experiments show that this labeling scheme improves the NDCG of two Web search rankers on several real-world test sets, with a low labeling overhead of around 1.4 labels per sample. This labeling scheme also outperforms several methods of using overlapping labels, such as simple k-overlap, majority vote, the highest labels, etc. Finally, the paper presents a study of how many overlapping labels are needed to get the best improvement in retrieval accuracy.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Storage and Retrieval.

General Terms

Algorithms, Experimentation.

Keywords

Relevance Labels, Overlapping Labels, Learning to rank.

1. INTRODUCTION

The retrieval accuracy of a learned model depends both on the quality of the training labels and on the amount of training examples. As expected, the higher the quality of the training labels, and the more the training examples, the better the accuracy of the learned model. A large set of training data is commonly used to improve a model's retrieval accuracy. Recently, however, researchers found that the improvement of the retrieval accuracy of a learned model stops after the number of training examples reaches a certain threshold [5]. When more training examples are not able to further improve a model's accuracy, improving the quality of labels is a solution.

Collecting high quality labels is a challenging task. Label quality depends both on the expertise of the labelers and on the number of labelers. For a given training sample, the more expert the labelers, and the more labelers, the higher the label quality. Therefore, the labels with the best quality should result from obtaining overlapping labels from multiple expert labelers.

However, obtaining overlapping labels from multiple experts is expensive. One alternative is to obtain overlapping labels from non-experts, for example, labelers from Amazon's Mechanical Turk (MTurk), which is an online labor market where workers are paid small amounts of money to complete human intelligence tasks. There is ongoing research [4] on how to effectively use services such as MTurk to obtain more labels. Unfortunately, labels from non-experts are often unreliable; and such unreliable labels decrease the retrieval accuracy of a learned model.

Another alternative to obtaining overlapping labels from experts is collecting one label from one expert for each sample. This is called the *single labeling scheme*. This single labeling scheme is affordable in general, and is widely used in supervised learning. However, since only one expert is involved in determining the relevance of a sample, the single expert's opinion may contain a personal bias, which may introduce noise, and thus possibly interfere with the learning of the ranking model. For example, the Web document at www.svmsolutions.com may be labeled as "Good" for the query "SVM" by a given expert, but the same expert may label the document www.svmlight.joachims.org as "Bad" if (s)he is not an expert in machine learning. Hence, the single labeling scheme may create unreliable labels because not every judge is an expert for every query. Figure 1 depicts agreements between aggregated overlapping labels and the ground truth, and agreement between the best judge and the ground truth, for 5 Web queries and 111 urls. Figure 1 shows that when the number of overlapping labels is greater than 5, the aggregated labels achieve better quality than even the "best" single judge does. This motivates the use of overlapping labels instead of a single expert judge.

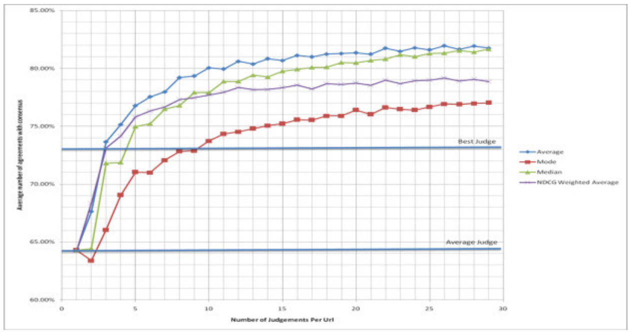


Figure 1: Aggregated Overlapping Labels are better than Best Judge.

SIGIR'10, July 19–23, 2010, Geneva, Switzerland.
Copyright 2010 ACM 978-1-60558-896-4/10/07...$10.00.

In this paper, we present a new scheme of collecting high quality labels at low cost. In particular, the paper focuses on how to cheaply and effectively produce and employ overlapping labels from multiple experts to improve Web search accuracy. The proposed labeling scheme requests additional labels from more experts only when a sample is labeled as relevant by one expert; otherwise, only the single label from that expert is used. Our experiments show that this selective labeling scheme improves Web search accuracy, which is measured in NDCG (Normalized Discounted Cumulative Gain [3]), of both LambdaRank [2] and LambdaMart [11] rankers on several real-world Web test sets, with a low labeling overhead of around 1.4 labels per sample. The proposed new labeling scheme also outperforms several methods of using overlapping labels, such as majority vote, k-overlap labels, the highest labels, etc. Furthermore, our paper also describes how many additional labels are needed to get the best improvement in retrieval accuracy.

Although this paper focuses on the task of *Web search*, the techniques presented can be generally applied to many research areas, such as computational linguistics, where manual labels are useful for training and evaluation.

Our paper makes two major contributions. First, the paper explores whether, when, and for which samples one should obtain overlapping training labels, as well as how many overlapping labels are needed. Second, the proposed *If-good-k* scheme creates high quality labels at low cost, which makes the approach promising for application to search engine training or other supervised training applications.

The remainder of the paper is organized as the follows. Section 2 describes the related work. Section 3 introduces the Web search task and the label distributions of multiple experts. Section 4 discusses issues of using overlapping labels. Section 5 details the proposed labeling scheme. Section 6 shows the experimental results and compares our approach to other commonly used overlapping labeling schemes. Section 7 discusses the proposed scheme, and Section 8 concludes the paper.

2. RELATED WORK

Due to the importance of search engine ranking and the advent of MTurk, research on expert labeling has recently become popular. In [9], a high agreement between MTurk non-expert annotations and existing gold-standard labels provided by expert labelers for five natural language processing tasks is demonstrated. Multiple labeling has also been shown to be useful for improving the data quality of annotations of non-experts [5].

Bernstein and Li pointed out in [1] that error-prone labelers often perform worse than a simple supervised learning setting using the initially labeled data. Their study suggested that the key to practical use of active learning with human labelers is to help the human labelers make fewer labeling mistakes.

Earlier work on inferring ground truth from subjective labels includes Smyth et al. [6], Sheng et al. [5], and Snow et al. [9]. In [6], the latent relation between subjective labels and true labels by EM algorithm was studied and it was shown that the posterior conditional probabilities of subjective labels and true labels generally agree with intuition and often (70% of the samples) correspond to a majority vote among the labelers. However, posterior conditional probabilities of subjective labels and true labels of the other 30% samples cannot be derived from the majority vote scheme.

The authors of [8] calculated a simple bound on the average classification accuracy across all labelers given the labels. The true labels are unknown. The bound is obtained by following the fact that the errors from all labelers are bounded by the maximum number of same-value labels from these labelers. This bound can be used to evaluate the quality of the overall labeling process.

The most common source of uncertainty of labels is subjective opinion, either from expert or non-expert labelers. In [7], it is mentioned that labeling of specific items may in itself be inconsistent, whether it is multiple labels from a single labeler at different times or labels from different labelers. In particular, in [7], Smyth et al. evaluated the labels created by two experts, who grouped samples into 5 label probability bins, against the ground truth, which are the consensus labels created by these two experts together. Smyth et al. also found that the labeling of individual experts relative to the consensus is not good.

Sheng et al. [5] pointed out that the improvement due to overlapping labels is more obvious when a single label is of low quality. However, their strategy requests a relatively large number of repeated labels for each sample. This high cost makes this approach impractical if the labeling cost for each labeler is high.

An interesting observation reported in [5] is that directly using multiple overlapping labels for each sample produces better label quality and better classification accuracy than using majority vote of the overlapping labels for each sample. In our experiments, we have similar findings on retrieval accuracy.

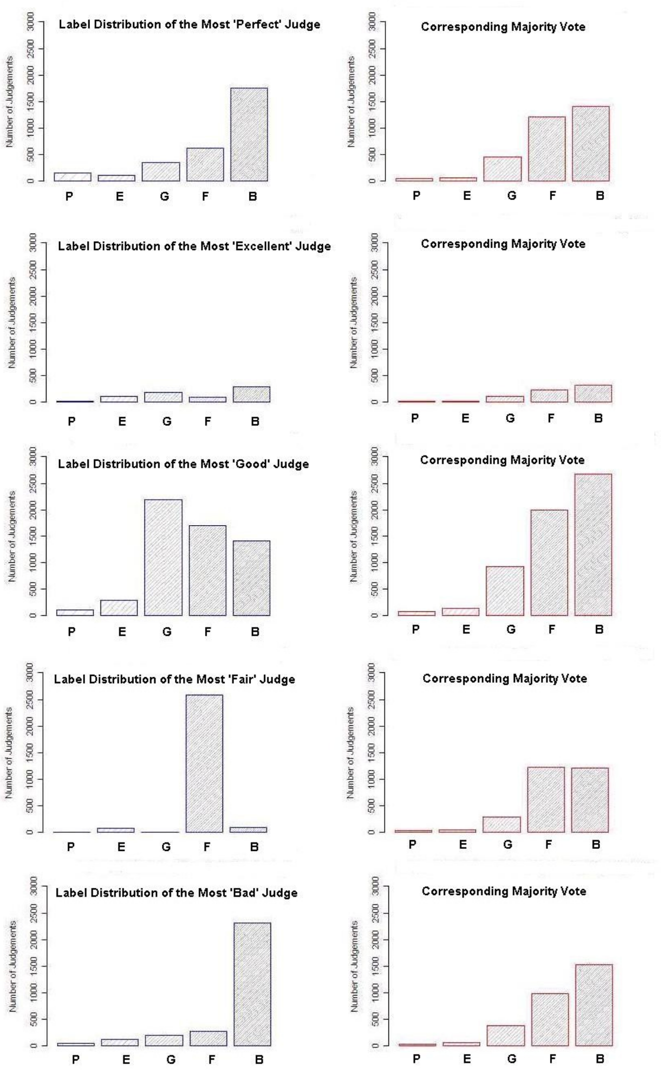
Moreover, not all samples need overlapping labels. In [5], Sheng et al. suggested using overlapping labels for samples whose overlapping labels show low agreement, and for samples whose overlapping labels bring high uncertainty to a learned model. Their method requires repeatedly labeling of each sample to determine whether using those overlapping labels for a sample in the training process. In this paper, we propose a more cost-effective method to select samples than to repeatedly label them.

The labeling scheme practically used in TREC-9 is perhaps the most similar settings to our proposed method. [10] describes how TREC-9 assessors judged the best page for a retrieval task. In their assessment, three different judges selected the best page(s) for a topic. The original assessor evaluated the entire pool for the topic and selected the best pages; the other two assessors were only given the documents the original assessor judged relevant or highly relevant. Remarkably, they also selected the best pages. The two secondary assessors did not know which were the best pages selected by the original assessor, but knew that all the given pages were judged relevant by the original assessor.

3. BACKGROUND

3.1 The Web Search Task

The Web search task takes in a set of queries and a set of retrieved Web documents for each query. Each Web document is represented by a feature vector consisting of features extracted from the anchor text, body content, url, title, and so on. A ranker learns a model from the training data, and computes a rank order of the urls based on their real-value relevance scores at the query level. In this paper, the two rankers used in the experiments are LambdaRank, a state-of-the-art neural network ranker, and



(a) (b)

Figure 2: Label Distributions of the Featured Judges and Label Distributions of the Corresponding Majority Vote on the Clean set (P=Perfect, E=Excellent, G=Good, F=Fair, B=Bad).

LambdaMART, a state-of-the-art ranker based on boosted regression trees and lambda-gradients.

The training data consists of label(s) and a feature vector for each query-url pair. A label indicates the relevance of a url to a query. Each query-url pair is judged on a 5-level relevance system scaling from highly relevant to not relevant. The 5 levels and respective numeric values are: *Perfect (4), Excellent (3), Good (2), Fair (1),* and *Bad (0)*.

3.2 Individual Opinion vs. Consensus

One of the label set used in our experiments is called Clean, which consists of 2,093 queries and 39,268 query-url pairs, with on average 19 urls per query. The queries in Clean were selected by hand from a large online query log.

There were 120 judges involved in the labeling process of Clean. Each query-url pair was judged by 11 judges. We found that the judges showed individual differences; some judges are more extreme and tend to label a sample either *Perfect* or *Bad*, whereas other judges are more moderate and tend to label a sample *Fair*. Figure 2(a) shows label distributions of the "featured" judges, who consistently assigned one label the most frequently among all the judges. For instance, "the most *Perfect* judge" assigns *"Perfect"* to samples the most frequently among all the judges. Comparing the label distributions of the five "featured" judges, we found that there exists statistically significant difference among individual judges' opinions.

Moreover, their individual opinions are also statistically significantly different from the consensus of all judges involved. Figure 2(b) shows the corresponding majority vote (consensus) among the judges who judged the same set of query-url pairs as the featured judge. Comparing the label distributions of the featured judges and the label distribution of the corresponding majority vote, we found that the individual opinions again are statistically significantly different from the consensus.

Since individual opinion differs from consensus, which one is better? Can we make use of overlapping labels to remove the individual variance and improve a search engine's retrieval accuracy? The variances among individual judges and the differences between an individual judge and consensus both post challenges to discovering an effective way to use overlapping labels. The following sections provide answers to these questions.

4. USING OVERLAPPING LABELS

As seen in Figure 1, multiple labelers can lead to cleaner higher quality training sets than a single best judge. This section discusses issues of using overlapping relevance labels. Section 4.1 presents methods of aggregating overlapping labels. Section 4.2 discusses the methodology of assigning different weights to different labels.

4.1 Aggregating Overlapping Labels

Suppose there are n samples, each of which needs to be labeled. For each sample, there are k labelers; each of them assigns one label to the sample, which yields k labels per sample.

There are many commonly used methods of aggregating k overlapping labels per sample. We focus on the following three widely used aggregation methods.

4.1.1 K-Overlap (Using All Labels)

This simple method is to train a model using all overlapping labels for each sample. That is, we input k labels (from k labelers) for each query-url pair, rather than one label. We call this method *k-overlap* since it uses k overlapping labels for each sample. The feature vector of a sample is repeatedly used for the k labels. Therefore for each training sample, there are k training instances with identical feature vectors and k labels each come from a different labeler.

The number of training instances is increased from n to kn. Hence, this method yields a training cost of kn. The labeling cost is k since k labelers are required for this scheme.

- Note that when k=1, we have the *single labeling scheme,* where each sample has exactly one label. This is the most commonly used labeling scheme in supervised learning, including learning to rank. The single labeling scheme yields a training cost of n, and a labeling cost of 1.

4.1.2 Majority Vote

Another commonly used method of aggregating overlapping labels is *majority vote*. The majority vote of multiple overlapping labels is the most frequent label among the k labels. If there is a tie (for example, there are 2 labels are *Good*, 2 Fair and 1 Bad, then there is a tie between *Good* and *Fair*), we first sort the most frequent labels in the order of most-relevant to least-relevant, i.e., in the order of *Perfect, Excellent, Good, Fair,* and *Bad*. For the above example, the most frequent labels are sorted into (*Good*, *Fair).* The majority vote is then picked as the label which is indexed by *ceiling(m/2)*, where *m* is the number of most frequent labels. In this case, *m*=2, *ceiling(m/2)*=1, which corresponds to *Good*. Here the index starts from 1.

This method yields a training cost of *n* (as low as the single labeling scheme) since each sample uses one label in the training process; and a labeling cost of *k* since it required *k* labels.

4.1.3 The Highest Labels

This aggregating scheme is designed in particular for the Web search task. Out of the *k* labels for a sample, the highest label is obtained by first sorting the *k* labels into the order of most-relevant to least-relevant, i.e., in the order of *Perfect, Excellent, Good, Fair,* and *Bad*; then picking the label at the top of the sorted list. For the above example, the highest label is *Good*.

This method yields a training cost of *n* (as low as the single labeling scheme) since each sample uses one label in the training process; and a labeling cost of *k* since it required *k* labels.

4.2 Weighting the Labels

Web search is a task which emphasizes precision more than recall. It suggests that finding a perfect relevant url to a query is much more important than finding all relevant urls to a query. It is particularly true when using an evaluation metric such as NDCG or Precision at rank position *n*, which are both common evaluation measures for search ranking models. Therefore, a sample which is labeled as *"Perfect"* probably deserves more weight during the model training process.

Moreover, relevant labels, such as *"Perfect"*, are rare and should be weighted more during the training process. LambdaRank and LambdaMart optimize directly for NDCG emphasize more on relevant labels and on higher positions in the ranking. In Figure 3, the label distribution of all of the labels in Clean shows that in the order of most-relevant to least-relevant, the amount of labels quadratically increases. The number of relevant samples is much less than the number of non-relevant samples.

In this paper, we assign different weights to labels in a simple way: for samples labeled as *Perfect/Excellent/Good*, we assign a training weight w_1, and for samples labeled as *Fair* or *Bad*, we assign a training weight w_2, where $w_1=\theta w_2$, and $\theta>1$.

The different weights of labels will neither change the training cost nor the labeling cost of the aggregating methods mentioned in Section 4.1, since both the number of training samples and the number of labels do not change.

5. THE SELECTIVE LABELING SCHEME

This section describes our proposed labeling scheme, *if-good-k*. In particular, the section studies how to cheaply and effectively use overlapping labels to improve a search engine's retrieval accuracy. The new scheme assigns additional labels to a subset of samples. Section 5.1 describes the intuition behind this scheme. Section 5.2 describes this selective labeling scheme.

5.1 The Intuition

People are difficult to satisfy, in particular in Web search, partly because of the gap between the real information need of a user and the query (s)he issues, and partly because of the limitations of the state-of-the-art retrieval technology. It is rare that a user finds that a url is perfectly relevant to his/her query, in particular for informational queries, rather (s)he often feels that a url is a bad result for his/her query. Therefore, if a labeler thinks a url is relevant to a given query, it is worthwhile to verify others' opinions, If a labeler thinks a url is bad, his/her opinion should be trusted.

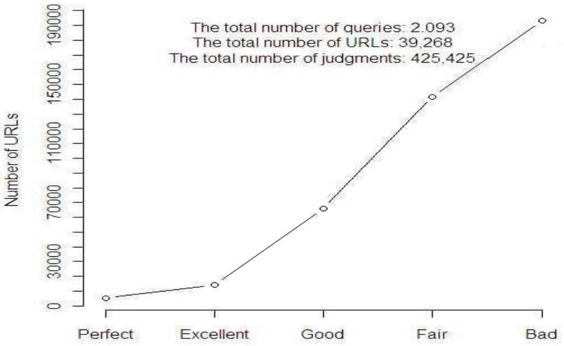


Figure 3: Label Distribution of the Clean Label Set.

Since there are relatively few highly relevant samples, training on highly relevant samples can be unstable. Moreover, it is usually hard for people to agree on some urls being relevant to a query. Labelers often disagree in their selection of the *Perfect* urls. This disagreement among labelers introduces variance and noise in the training data.

Based on this intuition, we propose to pay more attention to urls labeled as relevant rather than urls labeled as non-relevant. In our case, the relevant labels are *"Good and above" (Good+)*, i.e., *Perfect, Excellent,* or *Good*; and the non-relevant labels are *"Fair and below" (Fair-)*, i.e., *Fair* or *Bad*.

5.2 Selective Overlapping Labeling

5.2.1 If-Good-k

This scheme collects *k-1* additional overlapping labels only on samples previously judged as "*Good and above*", i.e., *Perfect, Excellent,* or *Good*. If a sample is judged as *"Fair and below"*, i.e., *Fair* or *Bad*, no additional label are requested and only the original label is used.

For example, when *k*=3, for a list of url samples, the following labels are given by three labelers under this scheme: *(Excellent, Good, Fair), (Bad), (Good, Good, Perfect), (Fair), (Fair), (Perfect, Fair, Good).*

The training cost and the labeling cost of this scheme depend on the Good+:Fair- ratio among the first labelers. If the Good+:Fair- ratio among the first labelers is *r*, then both the training and labeling cost of *if-good-k* are $\frac{n}{r+1}+\frac{nr}{r+1}k$.

5.2.2 Good-Till-Bad

This scheme continues to collect additional overlapping labels on samples previously judged as *"Good and above"* until the labels meet the first *"Fair and below"*. If a sample is judged as "Fair and below", i.e., *Fair* or *Bad*, no additional label are requested and the original label is used.

This scheme assumes a larger set of labelers are available, but not unlimited. There is still an upper bound of how many labels can be obtained for each sample. Suppose there are *k* labelers, then this scheme will generate up-to-*k* overlapping labels for a sample.

For example, when k=11, for a list of query-url pairs, the following labels are given by these 11 labelers under this scheme: *(Excellent, Good, Fair), (Bad), (Good, Good, Perfect, Excellent, Good, Bad), (Fair).*

The pros of *good-till-bad* over *if-good-k* is that it encourages collecting more additional labels for samples previously judged relevant, which provides more training data for samples that need more attentions. The cons of *good-till-bad,* comparing to *if-good-k,* is that it may result in a more expensive labeling cost since it requests a relatively larger pool of labels.

The training cost and the labeling cost of *good-till-bad* also depend on the Good+:Fair- ratio (the ratio of Good or better labels to Fair or worse labels) among the first labelers. If the Good+:Fair- ratio among the first labelers is r, and there are k labelers, then both the training and labeling cost of *good-till-bad* are *at most* $\frac{n}{r+1}+\frac{nr}{r+1}k$.

6. EXPERIMENTS

6.1 Datasets

There are two label sets used in the experiments. The first label set is called Clean and consists of 2,093 queries and 39,268 query-url pairs, with on average 19 urls per query. The queries in Clean were selected by hand from a large online query log. The labeling was done in December 2007. There were 120 judges involved in the labeling process of Clean. Each query-url pair was judged by 11 judges, hence there are 11 labels for each query-url pair. 364 queries do not have judgments from the same set of 11 judges for all urls for that query. In all other cases, all urls for a given query have been judged by the same set of 11 judges.

There are two feature sets used for Clean in our experiments. Note the two feature sets were generated in different periods, and differ in that the click and anchor information may have changed for the urls in the Clean label set, or that the pages themselves may have changed during the five months between the creation of the two feature sets. One set of features were obtained in August 2007. The training data consisting of these feature vectors and the Clean labels, called Clean07, contains 2,071 queries and 31,867 urls. The corresponding standard test set contains 5,207 queries and 930,951 urls. The second feature sets were obtained in January 2008. The training data created from these feature vectors and the Clean set labels, called Clean08, contains 1,563 queries and 287,903 urls. The corresponding standard test set contains 7,260 queries and 1,093,020 urls.

Another label set is called Clean+ and consists of 1,000 queries and 49,785 query-url pairs. Clean+ was created specifically to evaluate our proposed labeling scheme, *if-good-k*. The query-url pairs were labeled in a way that if the first judge labeled a pair as *Perfect*, *Excellent*, or *Good*, then two additional labels were requested from two additional judges, which yielded 17,800 additional labels; if the first judge labeled a query-url pair as *Fair* or *Bad*, then no more labels were requested. The feature vectors were obtained in July 2009 and the overlapping labels were collected during the week of 08/24/2009. The corresponding standard test set contains 11,898 queries, and 1,732,516 urls.

6.2 Evaluation Metrics

NDCG [3] is used as the evaluation metric in our experiments. NDCG is a retrieval measure which recognizes multilevel relevance labels. It is particularly suitable for Web search applications since it accounts for multilevel relevance and the truncation level can be set to model user behavior. NDCG for a given query at truncation level L is calculated as follows:

$$NDCG@L = \frac{1}{Z}\sum_{i=1}^{L}\frac{2^{l(i)}-1}{\log(1+i)} \qquad (1)$$

where $l(i) \in \{0,1,2,3,4\}$ is the relevance label of the document at rank position i, and L is the level to which NDCG is computed.

The main evaluation metric used in the experiments is NDCG@3. We also report NDCG@1, NDCG@5, and NDCG@10.

6.3 Experimental Settings

Based on Section 4 and Section 5, there are many different methods of using overlapping labels. Due to space limitations, we only report results for 9 interesting experimental settings. The 9 experimental settings are as follows:

Baseline: This is the single labeling scheme. There is one label for each query-url pair. For the Clean label set, the baseline labels are simulated by randomly drawing 1 label from the 11 labels. For Clean+, the labels from the first labelers are used as the baseline.

3-overlap: This is the *k-overlap* method with k=3. There are 3 overlapping labels obtained for each query-url pair. The rankers are trained on all overlapping labels and corresponding feature vectors. For the Clean label set, the 3 overlapping labels are created by randomly drawing 3 labels from the 11 labels. This experiment setting is not applicable to Clean+.

11-overlap: This is the *k-overlap* method with k=11.This experimental setting uses all 11 labels in the Clean label set as the training data. This setting is not applicable to Clean+.

Mv3: This is the majority vote method over 3 overlapping labels. The 3 overlapping labels are drawn randomly from the Clean label set. This experiment setting is not applicable to Clean+.

Mv11: This is the majority vote method over 11 overlapping labels. The 11 overlapping labels are all from the Clean label set. This setting is not applicable to Clean+.

If-good-3: This is the if-good-k labeling scheme, with k=3. The 3 overlapping labels for Clean are randomly drawn from the 11 labels for each query-url pair. This setting is applicable to both Clean and Clean+.

If-good-x3: This experimental setting combines the idea of selective labeling and weighting labels. If a label is "Good or above", the label is assigned a weight which is θ times of the weight of other labels. In this setting, θ=3. This setting is applicable to both Clean and Clean+.

Highest-3: This experimental setting uses the most relevant label of each query-url pair for training. In particular, this setting uses the highest label among k overlapping labels, with k=3. The 3 overlapping labels for Clean are randomly drawn from the 11 labels per sample. This setting is not applicable to Clean+.

Good-till-bad: This is the *Good-till-bad* labeling scheme. The upper limit of overlapping labels is k, with k=11 for the Clean label set. This setting is not applicable to Clean+.

Note that for the Clean label set, which contains 11 labels for each query-url pair, there is a constraint of k≤11. We performed a random sampling from these 11 labels if k<11. Due to the randomness of getting k labels when k<11, the averaged results of

Table 1: Retrieval Accuracy of Using Overlapping Labels (Clean07, LambdaRank).

Experiment	NDCG@3	NDCG@1	NDCG@5	NDCG@10
If-good-3	49.55%**	46.23%	51.81%	55.38%
11-overlap	49.30%	45.77%	51.60%	55.09%
Good-till-bad	49.22%	45.72%	51.73%	55.23%
Highest-3	49.16%	45.75%	51.49%	55.01%
3-overlap	49.00%	45.52%	51.51%	54.90%
If-good-x3	48.98%	45.25%	51.26%	54.82%
Mv3	48.87%	45.07%	51.36%	54.93%
Mv11	48.69%	45.25%	51.11%	54.58%
Baseline	48.60%	45.18%	51.02%	54.51%

Table 2: Retrieval Accuracy of Using Overlapping Labels (Clean08, LambdaRank).

Experiment	NDCG@3	NDCG@1	NDCG@5	NDCG@10
If-good-3	45.99%*	45.03%	47.53%	50.53%
Highest-3	45.97%*	44.87%	47.48%	50.43%
11-overlap	45.96%*	44.93%	47.57%	50.58%
Mv11	45.89%	44.97%	47.56%	50.58%
If-good-x3	45.80%	44.73%	47.40%	50.13%
3-overlap	45.78%	44.77%	47.54%	50.50%
Mv3	45.66%	44.83%	47.09%	49.83%
Good-till-bad	45.58%	44.88%	47.05%	49.86%
Baseline	45.53%	44.72%	46.93%	49.69%

Table 3: Retrieval Accuracy of Using Overlapping Labels (Clean07, LambdaMart).

Experiment	NDCG@3	NDCG@1	NDCG@5	NDCG@10
If-good-3	45.93%*	44.63%	47.65%	50.37%
3-overlap	45.91%*	44.70%	47.59%	50.35%
11-overlap	45.48%	44.31%	47.02%	49.97%
Mv11	45.42%	44.46%	47.16%	50.09%
If-good-x3	44.80%	43.78%	46.42%	49.26%
Highest-3	44.77%	43.52%	46.49%	49.44%
Mv3	44.45%	43.48%	46.11%	49.12%
Baseline	44.01%	42.96%	45.56%	48.30%

Table 4: Retrieval Accuracy of Using Overlapping Labels (Clean08, LambdaMart).

Experiment	NDCG@3	NDCG@1	NDCG@5	NDCG@10
If-good-3	45.93%*	44.64%	47.65%	50.38%
11-overlap	45.89%*	44.69%	47.60%	50.33%
Mv11	45.48%	44.30%	47.10%	49.95%
Highest-3	45.40%	44.49%	47.14%	50.07%
If-good-x3	44.79%	43.78%	46.40%	49.25%
3-overlap	44.76%	43.51%	46.48%	49.43%
Baseline	44.45%	43.47%	46.09%	49.10%
Mv3	44.02%	42.95%	45.54%	48.28%

Table 5: Retrieval Accuracy of Using Overlapping Labels (Clean+, LambdaRank).

Experiment	NDCG@3	NDCG@1	NDCG@5	NDCG@10
If-good-2	48.57%**	50.53%	48.56%	50.02%
If-good-3	48.41%	50.33%	48.48%	49.89%
Baseline	48.20%	50.32%	48.31%	49.65%
If-good-x3	48.16%	50.04%	48.18%	49.61%

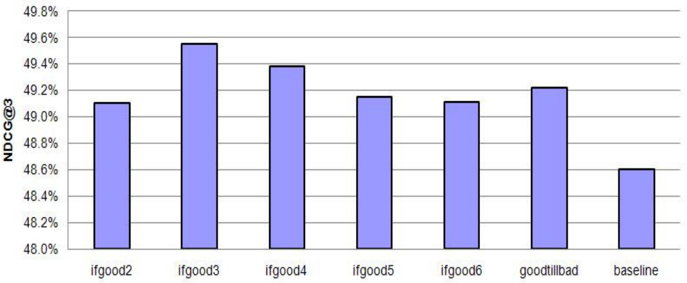


Figure 4: NDCG@3 for If-Good-k Runs (Clean07, LambdaRank).

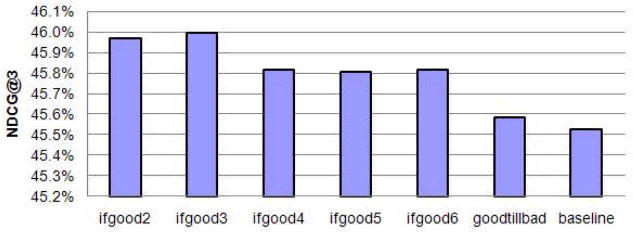


Figure 5: NDCG@3 for If-Good-k Runs (Clean08, LambdaRank).

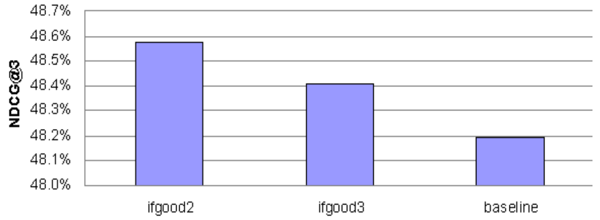


Figure 6: NDCG@3 for If-Good-k Runs (Clean+, LambdaRank).

5~10 runs are reported for each experimental setting. For the Clean+ label set, there is no such random sampling needed.

6.4 Effect on Retrieval Accuracy

This section reports search engine performance results for two supervised rankers. The two rankers used in the evaluation are LambdaRank and LambdaMart. Statistical significant tests are also performed to compare each experimental setting to the baseline. An NDCG@3 number is marked by "*" in the following tables means it is statistically significant better than the baseline in a t-test. An NDCG@3 number is marked by "**"means it is statistically significant better than all the other settings.

Table 1 shows the retrieval accuracy, which is measured in NDCG, for Clean07 using LambdaRank. The results are sorted decreasingly by NDCG@3. The best run is if-good-3, which has 0.95 point gain of NDCG@3 as compared to the baseline. This gain is statistically significant.

Table 2 shows the retrieval accuracy for Clean08 using LambdaRank. Similarly, the best run is *if-good-3*, which introduces 0.46 point gain of NDCG@3 as compared to the baseline. This gain is statistically significant.

Table 3 and Table 4 show the retrieval accuracy (measured in NDCG) when using LambdaMart trained on Clean07 and Clean08, respectively. The best runs for both experiments are *if-good-3*. As compared to the baseline, *if-good-3* introduces 1.92 point gain of NDCG@3 for Clean07 and 1.48 point gain of NDCG@3 for Clean08, respectively. The gains are statistically significant.

Table 5 shows the NDCG numbers of applying several selective overlapping labeling schemes and the baseline. LambdaRank is the ranker used in the evaluation. The results show that *if-good-2*, which requests only 1 additional overlapping label for samples originally labeled as Good+, yields the best NDCG numbers for NDCG@3, NDCG@1, NDCG@5, and NDCG@10. Based on NDCG@3, if-good-2 introduces a 0.37 point gain as compared to the baseline. This gain is statistically significant.

Note that although if-good-k (k=3 or k=2) runs consistently outperform all other methods and consistently statistically significantly outperform the baseline, the if-good-k runs may not always work statically significant better than some overlapping methods. For example, for Clean07 and LambdaMart, *if-good-3* produces similar retrieval accuracy as *3-overlap*. However, in Section 6.6 we will show that *if-good-k* is cheaper than other overlapping methods.

6.5 The Amount of Additional Labels

The experimental results reported in Section 6.4 show that selective overlapping labeling, in most cases, statistically significantly outperforms other methods of using overlapping labels. In the selective overlapping labeling scheme, for example, *if-good-k*, one may want to know the value of *k,* i.e., the number of additional labels required for the selected samples impact the ranking accuracy. In this section, we examine how many additional labels are needed for a selected sample to best improve the retrieval accuracy.

For the *if-good-k* scheme, *k-1* additional labels are requested from labelers. For the *good-till-bad* scheme, additional labels are requested from more labelers with an upper limit of 11 labels. The baseline is the single labeling scheme.

Figure 4 and Figure 5 show the NDCG@3 numbers of the *if-good-k* runs for Clean07 and Clean08, respectively. The tested *k* values range from 2 to 6. In addition, the *good-till-bad* scheme is also tested. Both schemes are compared with the baseline. The results indicate that when k=3, i.e., two additional labels are requested, the *if-good-3* scheme gives the best performance for both Clean07 and Clean08 datasets.

Figure 6 shows the NDCG@3 numbers of the *if-good-k* runs for Clean+. Since Clean+ only contains at most 3 labels for each query-url sample, the *k* values tested are 2 and 3. The best run is *if-good-2,* where only one additional label is requested for a selected sample which was previously labeled as Good+.

6.6 The Costs of Overlapping Labeling

The experimental results in Section 6.4 suggest that the proposed selective overlapping labeling scheme is promising to improve Web search retrieval accuracy. However, one may be concerned about the actual training and labeling costs associated with the proposed labeling scheme. The key reason for using our selective overlapping scheme is that it achieves similar and better accuracy to other methods, and significantly outperforms the baseline, with minimal labeling cost.

Table 6, Table 7, and Table 8 show the costs of various overlapping labeling schemes for Clean07, Clean08, and Clean+, respectively. In particular, these tables illustrate the labeling overhead, the training overhead, as well as the rate between the number of samples labeled as *Fair- (Fair or Bad)* and the number of samples labeled as *Good+ (Good, Excellent,* or *Perfect).*

Table 6: The Costs of Various Overlapping Labeling Schemes (Clean07).

Experiment	Labeling Overhead	Training Overhead	Fair-: Good+
Baseline	1	1	3.72
3-overlap	3	3	3.71
mv3	3	1	4.49
mv11	11	1	4.37
If-good-3	**1.41**	**1.41**	**2.24**
If-good-x3	1	1.41	2.24
Highest-3	3	1	1.78
Good-till-bad	1.87	1.87	1.38
11-overlap	11	11	4.37

Table 7: The Costs of Various Overlapping Labeling Schemes (Clean08).

Experiment	Labeling Overhead	Training Overhead	Fair-: Good+
Baseline	1	1	3.51
3-overlap	3	3	3.47
mv3	3	1	4.16
mv11	11	1	4.17
If-good-3	**1.45**	**1.45**	**2.09**
If-good-x3	1	1.45	2.09
Highest-3	3	1	1.58
Good-till-bad	1.98	1.98	1.23
11-overlap	11	11	4.17

Table 8: The Costs of Various Overlapping Labeling Schemes (Clean+).

Experiment	Labeling Overhead	Training Overhead	Fair-: Good+
Baseline	1	1	3.18
If-good-2	**1.23**	**1.23**	**2.09**
If-good-3	1.48	1.48	2.15
If-good-x3	1	1.48	1.06

The labeling overhead is calculated as the rate between the number of samples needed to be labeled by a labeling scheme and the number of sampled needed to be labeled by the baseline. The training overhead is calculated as the rate between the number of training samples by a labeling scheme and the number of training samples used by the baseline. The baselines are the single labeling scheme, where one label is assigned to each sample by one judge.

The runs which produce the best NDCG results in Section 6.4 are the *if-good-k* (*k*=2 or *k*=3) runs (in bold font in Tables 6-8). Tables 6-8 show that these *if-good-k* runs only generate a low labeling overhead of *around 1.4* as compared to the baselines. Such low labeling overhead suggests that the proposed overlapping labeling scheme is not only beneficial to improve retrieval accuracy, but also cost efficient and effective when used in ranking models.

7. DISCUSSIONS

The experiments show that the if-good-k labeling scheme not only improves the retrieval accuracy of both LambdaRank and LambdaMart on real-world Web test sets, but also consistently outperforms other methods of using overlapping labels in cost effectiveness, with NDCG gains significantly better than the baseline, and significantly better than other overlapping methods in most cases.

What is the secret behind the newly-proposed labeling scheme? At the beginning, we thought it is because more positive/relevant training samples (samples labeled as Good and above) are given to the ranker; hence the ranker has a more balanced training sample set. We therefore tried to simulate a training dataset with more Good+ samples by repeating a Good+ label for *k* times, which is equivalent to weighing a sample labeled as Good+ *k* times more than a sample labeled as Fair-. The *if-good*-x*3* runs reported in Table 1-5 are designed based on this. However, the results show that simply repeating the Good+ labels or simply increasing the weights of the Good+ samples do not work: the *if-good*-x*3* runs are ranked in the middle-to-low range of search performance among all the runs. Table 5 shows that in Clean+, the *if-good*-x*3* run even performs worse than the baseline. This shows that the performance gain of the selective overlapping labeling does not come from more Good+ samples.

We believe that the performance gain of *if-good-k* comes from generating higher quality labels. The *if-good-k* scheme correctly captures the worthiness of reconfirming a judgment for a sample. As we have mentioned in Section 5.1, if someone thinks a url is good, it is really worthwhile to double check with more people. On the other hand, if someone thinks a url is bad, we can trust the label. The *if-good-k* scheme yields higher quality labels by considering more opinions from different judges on those samples that need to be noise-free.

However, the experiments show that it is not true that the more the additional opinions for those selected samples, the better the retrieval accuracy. The experiments shown in Figures 4-6 were designed to find out how many additional labels are needed to most improve retrieval accuracy. It turns out that only a few additional labels are needed to improve the retrieval accuracy. In our experiments, one or two additional labels are good enough to beat other commonly used labeling schemes. It is not surprising since too many opinions from different labelers may create too much noise and too high variance in the training data; and it is hard for the model to learn useful information from such noisy and highly-variant training data.

In addition, the experiments in Section 6.4 suggest some labeling schemes do not work. It is surprising to see that the majority vote scheme performs significantly worse than just simply using all of the k labels *(K-overlap)* to train a model, in some cases. Moreover, as we mentioned earlier, simply changing weights for labels is not equivalent to collecting additional labels, *if-good*-x*k* does not perform as well as *if-good-k*.

The run which produces the best NDCG results with the lowest labeling cost for Clean07 and Clean08 is *if-good-3,* and for Clean+ is *if-good-2*. Note that these best runs only require a few additional labels. This makes both the labeling cost and the training cost for a good run low. As mentioned in Section 5.2, the costs of a selective overlapping labeling scheme depend on the label distribution of the original labels. Nevertheless, the *if-good-3* scheme is able to generate a labeling cost with a low overhead of around 1.4 labels per sample in general, which makes the proposed overlapping labeling an affordable and promising approach to be applied in real search engine training or other supervised training applications.

8. CONCLUSIONS

This paper explores whether, when, and for which samples one should obtain overlapping, expert training labels, as well as what to do with them once they have been obtained. In particular, this paper recommends a new method of effectively and efficiently producing and using overlapping labels to improve data quality and search engine's retrieval accuracy. The proposed selective labeling scheme requests additional labels only when the original labels are relevant. The experiments show that this labeling scheme improves the NDCG of both LambdaRank and LambdaMart ranking models on several real-world Web test sets, with a low labeling overhead of around 1.4 labels per sample. The proposed labeling scheme consistently outperforms several methods of using overlapping labels, such as majority vote, k-overlap labels, the highest labels, etc. Moreover, it is best to choose the labels via the *if-good-3* or *if-good-2* method, which achieves statistically significant NDCG@3 gain over using only one label per sample.

9. ACKNOWLEDGEMENT

We would like to thank Paul Bennett, Rich Caruana, and Chris Burges for their helpful discussions and feedback, Rangan Majumder for preparing the label sets, and anonymous reviewers for their useful comments and suggestions.

10. REFERENCES

[1] A. Bernstein and J. Li. From active towards interactive learning: Using consideration information to improve labeling correctness. University of Zurich, Dynamic and distributed information systems group working paper. www.ifi.uzh.ch/ddis/nc/publications.

[2] C. J. C. Burges, R. Ragno, and Q. V. Le. Learning to rank with nonsmooth cost functions. NIPS'06.

[3] Kalervo Järvelin and Jaana Kekäläinen. Ir evaluation methods for retrieving highly relevant documents. SIGIR'00.

[4] NAACL-HLT 2010 Workshop on Amazon Mechanical Turk. http://sites.google.com/site/amtworkshop2010/.

[5] V. S. Sheng, F. Provost, and P. G. Lpeirotis. Get another label? Improving data quality and data mining using multiple noisy labelers. KDD'08.

[6] P. Smyth, U. M. Fayyad, M. C. Burl, P. Perona, and P. Baldi. Inferring ground truth from subjective labeling of Venus images. NIPS'94.

[7] P. Smyth, M. C. Burl, U. M. Fayyad, P. Perona. Knowledge discovery in large image databases: Dealing with uncertainties in ground truth. AAAI Knowledge Discovery in Databases Workshop of KDD'94.

[8] P. Smyth. Bounds on the mean classification error rate of multiple experts. Pattern Recognition Letters 17, 12. 1996.

[9] R. Snow, B. O'Connor, D. Jurafsky and A. Y. Ng. 2008. Cheap and Fast – But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks. EMNLP'08.

[10] E. M. Voorhees. Evaluating by highly relevant documents. SIGIR'01.

[11] Q. Wu, C. J. C. Burges, K. M. Svore and J. Gao. Ranking, Boosting, and Model Adaptation. Microsoft Research Technical Report MSR-TR-2008-109.

[*] This work was done while the first author was an intern at Microsoft Bing/Microsoft Research.