Needle in a Haystack: Reducing the Costs of Annotating Rare-Class Instances in Imbalanced Datasets

Emily K. Jamison ‡ **and Iryna Gurevych** †‡
‡Ubiquitous Knowledge Processing Lab (UKP-TUDA),
Department of Computer Science, Technische Universität Darmstadt
† Ubiquitous Knowledge Processing Lab (UKP-DIPF),
German Institute for Educational Research
http://www.ukp.tu-darmstadt.de

Abstract

Crowdsourced data annotation is noisier than annotation from trained workers. Previous work has shown that redundant annotations can eliminate the agreement gap between crowdsource workers and trained workers. Redundant annotation is usually non-problematic because individual crowdsource judgments are inconsequentially cheap in a class-balanced dataset.

However, redundant annotation on class-imbalanced datasets requires many more labels per instance. In this paper, using three class-imbalanced corpora, we show that annotation redundancy for noise reduction is very expensive on a class-imbalanced dataset, and should be discarded for instances receiving a single common-class label. We also show that this simple technique produces annotations at approximately the same cost of a metadata-trained, supervised cascading machine classifier, or about 70% cheaper than 5-vote majority-vote aggregation.

1 Introduction

The advent of crowdsourcing as a cheap but noisy source for annotation labels has spurred the development of algorithms to maximize quality and minimize cost. Techniques can detect spammers (Oleson et al., 2011; Downs et al., 2010; Buchholz and Latorre, 2011), model worker quality and bias during label aggregation (Jung and Lease, 2012; Ipeirotis et al., 2010) and optimize obtaining more labels per instance or more labelled instances (Kumar and Lease, 2011; Sheng et al., 2008). However, much previous work for quality maximization and cost limitation assumes that the dataset to be annotated is class-balanced.

Class-imbalanced datasets, or datasets with differences in prior class probabilities, present a unique problem during corpus production: how to include enough rare-class instances in the corpus to train a machine learner? If the orginal class distribution is maintained, a corpus that is large enough for a machine learner to identify *common-class* (i.e., frequent class) instances may suffer from a lack of *rare-class* (i.e., infrequent class) instances. Yet, it can be cost-prohibitive to expand the corpus until enough rare-class instances are included.

Content-based instance targeting can be used to select instances with a high probability of being rare-class. For example, in a binary class annotation task identifying pairs of emails from the same thread, where most instances are negative, cosine text similarity between the emails can be used to identify pairs of emails that are likely to be positive, so that they could be annotated and included in the resulting class-balanced corpus (Jamison and Gurevych, 2013). However, this technique renders the corpus useless for experiments including token similarity (or ngram similarity, semantic similarity, stopword distribution similarity, keyword similarity, etc) as a feature; a machine learner would be likely to learn the very same features for classification that were used to identify the rare-class instances during corpus construction. Even worse, Mikros and Argiri (2007) showed that many features besides ngrams are significantly correlated with topic, including sentence and token length, readability measures, and word length distributions. The proposed targeted-instance corpus is unfit for experiments using sentence length similarity features, token length similarity features, etc.

Active Learning presents a similar problem of artificially limiting rare-class variety, by only identi-

28th Pacific Asia Conference on Language, Information and Computation pages 244–253

fying other potential rare-class instances for annotation that are very similar to the rare-class instances in the seed dataset. Rare-class instances may never be selected for labelling if they are very different from those in the seed dataset.

In this paper, we explore the use of cascading machine learner and cascading rule-based techniques for rare-class instance identification during corpus production. We avoid the use of content-based targeting, to maintain rare-class diversity, and instead focus on crowdsourcing practices and metadata. To the best of our knowledge, our work is the first work to evaluate cost-effective non-content-based annotation procedures for class-imbalanced datasets. Based on experiments with three class-imbalanced corpora, we show that redundancy for noise reduction is very expensive on a class-imbalanced dataset, and should be discarded for instances receiving a single common-class label. We also show that this simple technique produces annotations at approximately the same cost of a metadata-trained machine classifier, or about 70% cheaper than 5-vote majority-vote aggregation, and requires no training data, making it suitable for seed dataset production.

2 Previous Work

The rise of crowdsourcing has introduced promising new annotation strategies for corpus development.

Crowdsourced labels are extremely cheap. In a task where workers gave judgments rating a news headline for various emotions, Snow et al. (2008) collected 7000 judgments for a total of US$2. In a computer vision image labelling task, Sorokin and Forsyth (2008) collected 3861 labels for US$59; access to equivalent data from the annotation service *ImageParsing.com*, with an existing annotated dataset of 49,357 images, would have cost at least US$1000, or US$5000 for custom annotations.

Crowdsourced labels are also of usable quality. On a behavioral testing experiment of tool-use identification, Casler et al. (2013) compared the performance of crowdsource workers, social media-recruited workers, and in-person trained workers, and found that test results among the 3 groups were almost indistinguishable. Sprouse (2011) collected syntactic acceptability judgments from 176 trained undergraduate annotators and 176 crowdsource annotators, and after removing outlier work and ineligible workers, found no difference in statistical power or judgment distribution between the two groups. Nowak and Rüger (2010) compared annotations from experts and from crowdsource workers on an image labelling task, and they found that a single annotation set consisting of majority-vote aggregation of non-expert labels is comparable in quality to the expert annotation set. Snow et al. (2008) compared labels from trained annotators and crowdsource workers on five linguistic annotation tasks. They created an aggregated *meta-labeller* by averaging the labels of subsets of n non-expert annotations. Inter-annotator agreement between the non-expert meta-labeller and the expert labels ranged from .897 to 1.0 with n=10 on four of the tasks.

Sheng et al. (2008) showed that although a machine learner can learn from noisy labels, the number of needed instances is greatly reduced, and the quality of the annotation improved, with higher quality labels. To this end, much research aims to increase annotation quality while maintaining cost.

Annotation quality can be improved by removing unconscientious workers from the task. Oleson et al. (2011) screened spammers and provided worker training by embedding auto-selected *gold instances* (instances with high confidence labels) into the annotation task. Downs et al. (2010) identified 39% of unconscientious workers with a simple two-question qualifying task. Buchholz and Latorre (2011) examined cheating techniques associated with speech synthesis judgments, including workers who do not play the recordings, and found that cheating becomes more prevalent over time, if unchecked. They examined the statistical profile of cheaters and developed exclusion metrics.

Separate weighting of worker quality and bias during the aggregation of labels can produce higher quality annotations. Jung and Lease (2012) learned a worker's annotation quality from the sparse single-worker labels typical of a crowdsourcing annotation task, for improved weighting during label aggregation. In an image labelling task, Welinder and Perona (2010) estimated label uncertainty and worker ability, and derived an algorithm that seeks further labels from high quality annotators and controls the number of annotations per item to achieve a desired

level of confidence, with fewer total labels. Tarasov et al. (2014) dynamically estimated annotator reliability with regression using multi-armed bandits, in a system that is flexible to annotator unavailability, no gold standard, and a variety of label types. Dawid and Skene (1979) used an EM algorithm to simultaneously estimate worker bias and aggregate labels. Ipeirotis et al. (2010) separately calculated bias and error, enabling better quality assessment of a worker.

Some research explores the decision between obtaining more labels per instance or more labelled instances. Sheng et al. (2008) evaluated machine learning performance with different corpus sizes and label qualities. They evaluated four algorithms for use in deciding between redundant labelling and more labelled instances. Kumar and Lease (2011) built on the model by Sheng et al. (2008), adding knowledge of annotator quality for faster learning.

Other work focuses on correcting labels at the instance level. Dligach and Palmer (2011) used annotation-error detection and ambiguity detection to identify instances in need of additional annotations. Hsueh et al. (2009) modelled annotator quality and ambiguity rating to select highly informative yet unambiguous training instances.

Alternatively, class imbalance can be accommodated during machine learning, by resampling and cost-sensitive learning. Das et al. (2014) used density-based clustering to identify clusters in the instance space: if the clusters exceeded a threshold of majority-class dominance, they are undersampled to increase class-balance in the dataset. Batista et al. (2004) examined the effects of sampling for class-imbalance reduction on 13 datasets and found that oversampling is generally more effective than undersampling. They evaluated oversampling techniques to produce the fewest additional classifier rules. Elkan (2001) proved that class balance can be changed to set different misclassification penalties, although he observed this is ineffective with certain classifiers such as decision trees and Bayesian classifiers, so he also provided adjustment equations for use in such cases.

One option to reduce annotation costs is the classifier cascade. The Viola-Jones cascade machine learning-based framework (Viola and Jones, 2001) has been used to cheaply classify easy instances while passing along difficult instances for more costly classification. Classification of annotations can use annotation metadata: Zaidan and Callison-Burch (2011) used metadata crowdsource features to train a system to reject bad translations in a translation generation task. Cascaded classifiers are used by Bourdev and Brandt (2005) for object detection in images and Raykar et al. (2010) to reduce the cost of obtaining expensive (in money or pain to the patient) features in a medical diagnosis setting. In this paper, we evaluate the use of metadata-based classifier cascade, as well as rule cascades, to reduce annotation costs.

3 Three Class-Imbalanced Annotation Tasks

We investigate three class-imbalanced annotation tasks; all are pairwise classification tasks that are class-imbalanced due to factorial combination of text pairs.

Pairwise Email Thread Disentanglement A pairwise email disentanglement task labels pairs of emails with whether or not the two emails come from the same email thread (a *positive* or *negative* instance). The Emails dataset[1] consists of 34 positive and 66 negative instances, and simulates a server's contents in which most pairs are negative (common class). The emails come from the Enron Email Corpus , which has no inherent header thread labelling. Annotators were shown both texts side-by-side and asked "Are these two emails from the same discussion/email thread?" Possible answers were *yes*, *can't tell*, and *no*.

Pairwise Wikipedia Discussion Turn/Edit Alignment Wikipedia editors discuss plans for *edits* in an article's *discussion* page, but there is no inherent mechanism to connect specific *discussion turns* in the discussion to the edits they describe. A corpus of matched turn/edit pairs permits investigation of relations between turns and edits. The Wiki dataset[2] consists of 750 turn/edit pairs. Additional rare-class (positive) instances were added to the corpus, resulting in 17% positive instances. Annotators were

[1] www.ukp.tu-darmstadt.de/data/text-similarity/email-disentanglement/

[2] www.ukp.tu-darmstadt.de/data/discourse-analysis/wikipedia-edit-turn-pair-corpus/

Sentence1: *Cord is strong, thick string.*
Sentence2: *A smile is the expression that you have on your face when you are pleased or amused, or when you are being friendly.*

Figure 1: Sample text pair from text similarity corpus, classified by 7 out of 10 workers as 1 on a scale of 1-5.

shown the article topic, turn and thread topic, the edit, and the edit comment, and asked, "Does the Wiki comment match the Wiki edit?" Possible answers were *yes*, *can't tell*, and *no*.

Sentence Pair Text Similarity Ratings To rate sentence similarity, annotators read 2 sentences and answered the question, "How close do these sentences come to meaning the same thing?" Annotators rated text similarity of the sentences on a scale of 1 (minimum similarity) to 5 (maximum similarity). This crowdsource dataset was produced by Bär et al. (2011). An example sentence pair is shown in Figure 1. The SentPairs dataset consists of 30 sentence pairs.

The original classification was calculated as the mean of a pair's judgments. However, on a theoretical level, it is unclear that mean, even with a deviation measure, accurately expresses annotator judgments for this task. Our experiments (see Sections 6 and 7) use mode score as the gold standard, which occasionally results in multiple instances derived from one set of ratings.

From the view of binary classification, each one of the 5 classes constitutes a rare class. For the purposes of our experiments, we treat each class in turn as the rare-class, while neighboring classes are treated as *can't tell* classes (with estimated normalization for continuum edge classes *1* and *5*), and the rest as common classes. For example, experiments treating class *4* as rare treated classes *3* and *5* as "*can't tell*" and classes *1* and *2* as common.

4 How severe is class imbalance?

The Emails and Wiki datasets consist of two texts paired in such a way that a complete dataset would consist of all possible pair combinations (Cartesian product). Although the dataset for text similarity rating does not require such pairing, it is still heavily class imbalanced.

Consider an email corpus with a set of threads T and each $t \in T$ consisting of a set of emails E_t, where rare-class instances are pairs of emails from the same thread, and common-class instances are pairs of emails from different threads. We have the following number of rare-class instances:

$$|\text{ Instances }_{\text{rare}}| = \sum_{i=1}^{|T|} \sum_{j=1}^{|E_i|-1} j$$

and number of common-class instances:

$$|\text{ Instances }_{\text{common}}| = \sum_{i=1}^{|T|} \sum_{j=1}^{|E_i|} \sum_{k=(i+1)}^{|T|} |E_k|$$

For example, in an email corpus with 2 threads of 2 emails each, 4 (67%) of pairs are common-class instances, and 2 (33%) are rare-class instances. If another email thread of two emails is added, 12 (80%) of the pairs are common-class instances, and 3 (20%) are rare-class instances.

To provide a constant value for the purposes of this work, we standardize rare-class frequency to 0.01 unless otherwise noted. This is different from our datasets' actual class imbalances, but the conclusions from our experiments in Section 7 are independent of class balance.

5 Baseline Cost

The baseline aggregation technique in our experiments (see Sections 6 and 7) is majority vote of the annotators. For example, if an instance receives at least 3 out of 5 rare-class annotations, then the baseline consensus declares it rare-class.

Emails Dataset Cost For our Emails dataset, we solicited 10 Amazon Mechnical Turk (*MTurk*)[3] annotations for each of 100 pairs of emails, at a cost of US$0.033[4] per annotation. Standard quality measures employed to reduce spam annotations included over 2000 *HIT*s (MTurk tasks) completed, 95% HIT acceptance rate, and location in the US.

Assuming 0.01 rare-class frequency[5] and 5 annotations[6], the cost of a rare-class instance is:

$$\frac{US\$0.033 \times 5 \text{ annotators}}{0.01 \text{ freq}} = US\$16.50$$

[3] www.mturk.com

[4] Including approx. 10% MTurk fees

[5] Although this paper proposes a hypothetical 0.01 rare-class frequency, the Emails and Wiki datasets have been partially balanced: the negative instances merely functioned as a distractor for annotators, and conclusions drawn from the rule cascade experiments only apply to positive instances.

[6] On this dataset, IAA was high and 10 annotations was over-redundant.

Wiki Dataset Cost For our Wiki dataset, we solicited five MTurk annotations for each of 750 turn/edit text pairs at a cost of US$0.044 per annotation. Measures for Wikipedia turn/edit pairs included 2000 HITs completed, 97% acceptance rate, age over 18, and either preapproval based on good work on pilot studies or a high score on a qualification test of sample pairs. The cost of a rare-class instance is:

$$\frac{US\$0.044 \times 5 \text{ annotators}}{0.01 \text{ freq}} = US\$22$$

SentPairs Dataset Cost The SentPairs datset consists of 30 sentence pairs, and 10 annotations per pair. The original price of Bär et al. (2011)'s sentence pairs corpus is unknown, so we estimated a cost of US$0.01 per annotation. The annotations came from Crowdflower[7]. Bär et al. (2011) used a number of quality assurance mechanisms, such as worker reliability and annotation correlation. The cost of a rare-class instance varied between classes, due to class frequency variation, from instance$_{class2}$=US$0.027 to instance$_{class5}$=US$0.227.

Finding versus Confirming a Rare-Class Instance It is cheaper to confirm a rare-class instance than to find a suspected rare-class instance in the first place. We have two types of binary decisions: finding a suspected rare-class instance ("Is the instance a true positive (*TP*) or false negative (*FN*)?") and confirming a rare-class instance as rare ("Is the instance a TP or false positive (*FP*)?"). Assuming a 0.01 rare-class frequency, 5-annotation majority-vote decision, and 0.5 FP frequency, the cost of the former is:

$$\frac{1 \text{ annotation}}{0.01 \text{ freq}} + \frac{1 \text{ annotation}}{0.99 \text{ freq}} = 101 \text{ annotations}$$

and the latter is:

$$\frac{5 \text{ annotations}}{0.5 \text{ freq}} = 10 \text{ annotations}$$

Metrics We used the following metrics for our experiment results:

TP is the number of true positives (rare-class) discovered. The fewer TP's discovered, the less likely the resulting corpus will represent the original data in an undistorted manner.

P$_{rare}$ is the precision over rare instances: $\frac{TP}{TP+FP}$ Lower precision means lower confidence in the produced dataset, because the "rare" instances we found might have been misclassified.

[7] crowdflower.com

AvgA is the average number of annotations needed for the system to label an instance common-class.

The normalized cost is the estimated cost of acquiring a rare instance: $\frac{\text{AvgA} \times \text{annoCost}}{\frac{\text{classImbalance}}{\text{Recall}_{rare}}}$

Savings is the estimated cost saved when identifying rare instances, over the baseline. Includes Standard Deviation.

6 Supervised Cascading Classifier Experiments

Previous work (Zaidan and Callison-Burch, 2011) used machine learners to predict which instances to annotate based on annotation metadata. In this section, we used crowdsourcing annotation metadata (such as time duration) as features for a cascading logistic regression classifier to choose whether or not an additional annotation is needed. In each of the five cascade rounds, an instance was classified as either *potentially rare* or *common*. Instances classified as potentially rare received another annotation and continued through the next cascade, while instances classified as common were discarded. Discarding instances before the end of the cascade can reduce the total number of needed annotations, and therefore lower the total cost. This cascade models the observation (see Section 5) that it is cheap to confirm suspected rare-class instances, but it is expensive to weed out common-class instances.

Experiments from this section will be compared in Section 7 to a rule-based cascading classifier system that, unlike this supervised system, does not need any training data.

6.1 Instances

Each experimental instance consisted of features derived from the metadata of one or more crowdsourced annotations from a pair of texts. A gold standard rare instance has >80% rare annotations.

In the first round of experiments, each instance was derived from a single annotation. In each further round, instances were only included that consisted of an instance from the previous round that had been classified *potentially rare* plus one additional annotation. All possible instances were used that could be derived from the available annotations, as long as the instance was permitted by the previous round of classification (see Figure 2). This maximized the

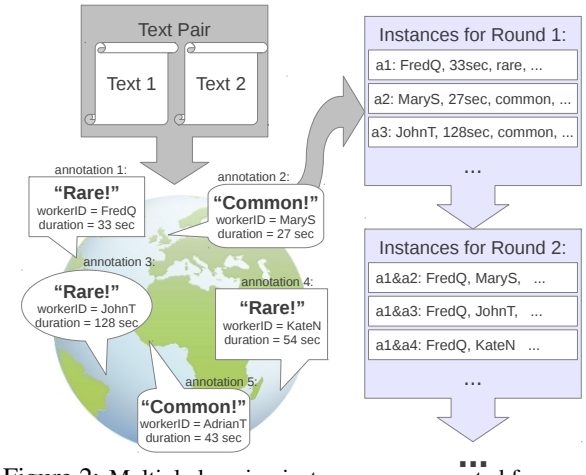

Figure 2: Multiple learning instances are generated from each original annotated text pair.

number of instances available for the experiments. K-fold cross-validation was used, but to avoid information leak, no test data was classified using a model trained on any instances generated from the same original text pairs.

Although SentPairs had 10 annotations per pair, we stopped the cascade at five iterations, because the number of rare-class instances was too small to continue. This resulted in a larger number of final instances than actual sentence pairs.

6.2 Features

Features were derived from the metadata of annotations. Features included an annotation's worker ID, estimated time duration, annotation day of the week (Emails and Wiki only), and the label (*rare*, *common*, *can't tell*), as well as all possible joins of one annotation's features (`commonANDJohnTAND30sec`). For instances representing more than a single annotation, a feature's *count* over all the annotations was also included (i.e., `common:3` for an instance including 3 *common* annotations). For reasons discussed in Section 1, we exclude features based on text content of the pair.

6.3 Results

Tables 1 and 2 show the results of our trained cascading system on Emails and Wiki, respectively; baseline is majority voting. Tables 3 and 4 show results on rare classes `1` and `5` of SentPairs (classes `2`, `3`, and `4` had too few instances to train, a disadvantage of a supervised system that is fixed by our rule-based system in Section 7); baseline is mode class voting.

features	TPs	P_{rare}	AvgA	Norm cost	Savings(%)
baseline	34	1.00	-	$16.50	-
anno	31	0.88	1.2341	$4.68	72±8
worker	0	0.0	1.0	-	-
dur	2	0.1	1.0	$16.5	0±0
day	0	0.0	1.0	-	-
worker & anno	33	0.9	1.1953	$4.38	73±7
day & anno	31	0.88	1.2347	$4.68	72±8
dur & anno	33	0.88	1.2437	$4.56	72±8
w/o anno	3	0.12	1.2577	$20.75	-26±41
w/o worker	33	0.9	1.2341	$4.53	73±8
w/o day	33	0.9	1.2098	$4.44	73±7
w/o dur	33	0.9	1.187	**$4.35**	**74±7**
all	33	0.9	1.2205	$4.48	73±8

Table 1: Email results on the trained cascade.

features	TPs	P_{rare}	AvgA	Norm cost	Savings(%)
baseline	128	1.00	-	$22.00	-
anno	35	0.93	1.7982	$20.29	08±32
worker	0	0.0	1.0	-	-
dur	0	0.0	1.0	-	-
day	0	0.0	1.0	-	-
worker & anno	126	0.99	1.6022	**$7.12**	**68±11**
day & anno	108	0.88	1.644	$8.51	61±13
dur & anno	111	0.86	1.5978	$8.08	63±12
w/o anno	4	0.12	1.0259	$11.28	49±6
w/o worker	92	0.84	1.7193	$9.46	57±15
w/o day	104	0.9	1.6639	$8.61	61±14
w/o dur	109	0.94	1.6578	$8.2	63±14
all	89	0.82	1.6717	$8.76	60±15

Table 2: Wiki results on the trained cascade.

Table 1 shows that the best feature combination for identifying rare email pairs was annotation, worker ID, and day of the week ($4.35 per rare instance, and 33/34 instances found); however, this was only marginally better than using annotation alone ($4.68, 31/34 instances found). The best feature combination resulted in a 74% cost savings over the conventional 5-annotation baseline.

Table 2 shows that the best feature combination for identifying rare wiki pairs was annotation and worker ID ($7.12, 126/128 instances found). Unlike the email experiments, this combination was remarkably more effective than annotations alone ($20.29, 35/128 instances found), and produced a 68% total cost savings.

Tables 3 and 4 show that the best feature combination for identifying rare sentence pairs for both rare classes 1 and 5 was also annotation and worker

features	TPs	P_{rare}	AvgA	Norm cost	Savings(%)
baseline	12	1.00	-	$1.50	-
anno	9	0.67	1.8663	$0.4	73±10
workerID	1	0.1	1.5426	$2.31	-54±59
dur	2	0.15	1.4759	$1.11	26±26
worker & anno	11	0.7	1.8216	**$0.39**	**74±9**
worker & dur	3	0.2	1.8813	$1.41	06±34
dur & anno	8	0.42	1.8783	$0.56	62±13
all	11	0.62	1.8947	$0.41	73±8

Table 3: SentPairs_{c1} results on the trained cascade.

features	TPs	P_{rare}	AvgA	Norm cost	Savings(%)
baseline	17	1.00	-	\$0.44	-
anno	14	0.72	2.4545	\$0.15	66±7
worker	14	0.63	2.7937	\$0.16	64±8
dur	10	0.52	2.7111	\$0.18	58±11
worker & anno	15	0.82	2.3478	**\$0.12**	**73±8**
worker & dur	6	0.4	2.7576	\$0.38	14±23
dur & anno	16	0.72	2.4887	\$0.14	69±10
all	17	0.82	2.4408	\$0.12	73±5

Table 4: SentPairs_{c5} results on the trained cascade.

ID (US\$0.39 and US\$0.12, respectively), which produced a 73% cost savings; for class 5, adding duration minimally decreased the standard deviation. Annotation and worker ID were only marginally better than annotation alone for class 1.

7 Rule-based Cascade Experiments

Although the meta-data-trained cascading classifier system is effective in reducing the needed number of annotations, it is not useful in the initial stage of annotation, when there is no training data. In these experiments, we evaluate a rule-based cascade in place of our previous trained classifier. The rule-based cascade functions similarly to the trained classifier cascade except that a single rule replaces each classification. Five cascades are used.

Each rule instructs when to discard an instance from further annotation. For example, `no>2` means, "if the count of *no* (i.e., common) annotations becomes greater than 2, we assume the instance is common and do not seek further confirmation from more annotations." A gold standard rare instances has >80% rare annotations.

For our rule-based experiments, we define AvgA for each instance i and for annotations a_{1_i}, a_{2_i}, ..., a_{5_i} and the probability (Pr) of five non-common-class annotations. Class c is the common class. We always need a first annotation: $\Pr(a_{1_i} \neq c) = 1$.

$$AvgA_i = \sum_{j=1}^{5} \prod_{k=1}^{j} \Pr(a_{k_i} \neq c)$$

We define *Precision*$_{rare}$ (P_{rare}) as the probability that instance i with 5 common[8] annotations a_{1_i}, a_{2_i}, ..., a_{5_i} is not a rare-class instance:

$$\begin{aligned} P_{rare_i} &= \Pr(\text{TP}|(a_{1\ldots5_i} = \text{rare})) \\ &= 1 - \Pr(\text{FP}|(a_{1\ldots5_i} = \text{rare})) \end{aligned}$$

Thus, we estimate the probability of seeing other FPs based on the class distribution of our annotations. This is different from our supervised cascade experiments, in which $P_{rare} = \frac{TP}{TP+FP}$.

[8]This may also include *can't tell* annotations, depending on the experiment.

7.1 Results

Table 5 shows the results of various rule systems on reducing cost on the wiki data.

While it might appear reasonable to allow one or two careless crowdsource annotations before discarding an instance, the tables show just how costly this allowance is: each permitted extra annotation (i.e., `no>1`, `no>2`, ...) must be applied systematically to each instance (because we do not know which annotations are careless and which are accurate) and can increase the average number of annotations needed to discard a common instance by over 1. The practice also decreases rare-class precision, within an *n*-annotations limit. Clearly the cheapest and most precise option is to discard an instance as soon as there is a common-class annotation.

When inherently ambiguous instances are shifted from rare to common by including *can't tell* as a common annotation, the cost of a rare Wiki instance falls from US\$7.09 (68% savings over baseline) to US\$6.10 (72% savings), and the best performing rule is `(no+ct)>0`. A rare email instance barely increases from US\$3.52 (79% savings) to US\$3.65 (78% savings). However, in both cases, TP of rare-class instances falls (Wiki: 39 instances to 22, Emails: 32 instances to 30). This does not affect overall cost, because it is already included in the equation, but the rare-class instances found may not be representative of the data.

There was not much change in precision in the Wiki dataset when *can't tell* was included as a rare annotation (such as `no>0`) or a common annotation (such as `(no+ct)>0`), so we assume that the populations of rare instances gathered are not different between the two. However, when a reduced number of TPs are produced from treating *can't tell* as a common annotation, higher annotation costs can result (such as Table 5, `no>0` cost of US\$7.09, versus `(no+ct)>0` cost of US\$10.56).

Removing ambiguous instances from the test corpus does not notably change the results (see Table 6). Ambiguous instances were those where the majority class was *can't tell*, the majority class was tied with *can't tell*, or there was a tie between common and rare classes.

Finally, the tables show that not only do the top-performing rules save money over the 5-annotations

Class = N if:	TP	P_{rare}	AvgA	NormCost	Savings(%)
baseline	128	1.00	-	$22.0	-
no $>$ 0	39	0.95	1.61	**$7.09**	**68±16**
no $>$ 1	39	0.85	2.86	$12.6	43±19
no $>$ 2	39	0.73	3.81	$16.75	24±15
(no+ct) $>$ 0	22	0.98	1.35	$10.56	52±20
(no+ct) $>$ 1	33	0.93	2.55	$13.25	40±18
(no+ct) $>$ 2	35	0.85	3.56	$17.44	21±15

Table 5: Wiki results: rule-based cascade. All instances included.

Class = N if:	TP	P_{rare}	AvgA	NormCost	Savings(%)
baseline	128	1.00	-	$22.0	-
no $>$ 0	35	0.96	1.46	**$6.43**	**71±14**
no $>$ 1	35	0.9	2.67	$11.76	47±17
no $>$ 2	35	0.81	3.66	$16.11	27±14
(no+ct) $>$ 0	22	0.98	1.33	$9.34	58±19
(no+ct) $>$ 1	33	0.92	2.5	$11.66	47±17
(no+ct) $>$ 2	35	0.85	3.49	$15.36	30±13

Table 6: Wiki results: no ambiguous instances.

baseline, they save about as much money as supervised cascade classification.

Table 7 shows results from the Emails dataset. Results largely mirrored those of the Wiki dataset, except that there was higher inter-annotator agreement on the email pairs which reduced annotation costs. We also found that, similarly to the Wiki experiments, weeding out uncertain examples did not notably change the results.

Results of the rule-based cascade on SentPairs are shown in Tables 8, 9, 10, and 11. Note there were no instances with a mode gold classification of 3. Also, there are more total rare instances than sentence pairs, because of the method used to identified a gold instance: annotations neighboring the rare class were ignored, and an instance was gold rare if the count of rare annotations was $>$0.8 of total annotations. Thus, an instance with the count {class1=5, class2=4, class3=1, class4=0, class5=0} counts as a gold instance of both class 1 and class 2.

The cheapest rule was `no>0`, which had a recall of 1.0, P_{rare} of 0.9895, and a cost savings of 80-83% (across classes 1-5) over the 10 annotators originally used in this task.

Class = N if:	TP	P_{rare}	AvgA	NormCost	Savings(%)
baseline	34	1.00	-	$16.5	-
no $>$ 0	32	1.0	1.07	**$3.52**	**79±6**
no $>$ 1	32	0.99	2.11	$6.95	58±7
no $>$ 2	32	0.98	3.12	$10.31	38±6
(no+ct) $>$ 0	30	1.0	1.04	$3.67	78±5
(no+ct) $>$ 1	32	0.99	2.07	$6.83	59±6
(no+ct) $>$ 2	32	0.99	3.08	$10.16	38±5

Table 7: Email results: rule-based cascade.

Class = N if:	TP	P_{rare}	AvgA	NormCost	Savings(%)
baseline	5	1.00	-	$1.5	-
no $>$ 0	5	0.99	1.69	**$0.25**	**83±10**
no $>$ 1	5	0.96	3.27	$0.49	67±17
no $>$ 2	5	0.9	4.66	$0.7	53±21
(no+ct) $>$ 0	0	1.0	1.34	-	-
(no+ct) $>$ 1	2	0.98	2.63	$0.98	34±31
(no+ct) $>$ 2	4	0.96	3.83	$0.72	52±19

Table 8: SentPairs_{c1} results: rule-based cascade.

Class = N if:	TP	P_{rare}	AvgA	NormCost	Savings(%)
baseline	2	1.00	-	$3.75	-
no $>$ 0	2	0.98	1.95	**$0.73**	**81±12**
no $>$ 1	2	0.93	3.68	$1.38	63±20
no $>$ 2	2	0.86	5.12	$1.92	49±23
(no+ct) $>$ 0	0	1.0	1.1	-	-
(no+ct) $>$ 1	0	1.0	2.2	-	-
(no+ct) $>$ 2	0	1.0	3.29	-	-

Table 9: SentPairs_{c2} results: rule-based cascade.

7.2 Error Analysis

A rare-class instance with many common annotations has a greater chance of being labelled common-class and thus discarded by a single crowdsource worker screening the data. What are the traits of rare-class instances at high risk of being discarded? We analyzed only Wiki text pairs, because the inter-annotator agreement was low enough to cause false negatives. The small size of SentPairs and the high inter-annotator agreement of Emails prevented analysis.

Wiki data The numbers of instances (750 total) with various crowdsource annotation distributions are shown in Table 12. The table shows annotation distributions (i.e., `302` = 3 *yes*, 0 *can't tell* and 2 *no*) for rare-class instance numbers with high and low probabilities of being missed.

We analyzed the instances from the category most likely to be missed (`302`) and compared it with the two categories least likely to be missed (`500`, `410`). Of five random `302` pairs, all five appeared highly ambiguous and difficult to annotate; they were missing context that was known (or assumed to be known) by the original participants. Two of the turns state future deletion operations, and the ed-

Class = N if:	TP	P_{rare}	AvgA	NormCost	Savings(%)
baseline	16	1.00	-	$0.47	-
no $>$ 0	16	0.99	1.98	**$0.09**	**80±9**
no $>$ 1	16	0.96	3.83	$0.18	62±15
no $>$ 2	16	0.9	5.47	$0.26	45±17
(no+ct) $>$ 0	0	1.0	1.23	-	-
(no+ct) $>$ 1	0	1.0	2.45	-	-
(no+ct) $>$ 2	1	0.99	3.65	$2.74	-484±162

Table 10: SentPairs_{c4} results: rule-based cascade.

Class = N if:	TP	P_{rare}	AvgA	NormCost	Savings(%)
baseline	17	1.00	-	\$0.44	-
no $>$ 0	17	0.99	1.96	**\$0.09**	**80±10**
no $>$ 1	17	0.95	3.77	\$0.17	62±16
no $>$ 2	17	0.89	5.37	\$0.24	46±18
(no+ct) $>$ 0	2	1.0	1.27	\$0.48	-8±21
(no+ct) $>$ 1	10	1.0	2.54	\$0.19	57±8
(no+ct) $>$ 2	13	1.0	3.8	\$0.22	50±9

Table 11: SentPairs$_{c5}$ results: rule-based cascade.

Ambiguous instances		Unambiguous instances	
Anno, *y ct n*	# inst	Anno, *y ct n*	# inst
3 0 2	35	5 0 0	22
3 1 1	30	4 1 0	11
2 2 1	19	4 0 1	28
2 1 2	39	3 2 0	2

Table 12: Anno. distributions and instance counts.

its include deleted statements, but it is unknown if the turns were referring to these particular deleted statements or to others. In another instance, the turn argues that a contentious research question has been answered and that the user will edit the article accordingly, but it is unclear in which direction the user intended to edit the article. In another instance, the turn requests the expansion of an article section, and the edit is an added reference to that section. In the last pair, the turn gives a quote from the article and requests a source, and the edit adds a source to the quoted part of the article, but the source clearly refers to just one part of the quote.

In contrast, we found four of the five `500` and `410` pairs to be clear rare-class instances. Turns quoted text from the article that matched actions in the edits. In the fifth pair, a `500` instance, the edit was first made, then the turn was submitted complaining about the edit and asking it to be reversed. This was a failure by the annotators to follow the directions included with the task, of which types of pairs are positive instances and which are not.

8 Conclusion

Crowdsourcing is a cheap but noisy source of annotation labels, encouraging redundant labelling. However, redundant annotation on class-imbalanced datasets requires many more labels per instance. In this paper, using three class-imbalanced corpora, we have shown that annotation redundancy for noise reduction is expensive on a class-imbalanced dataset, and should be discarded for instances receiving a single common-class label. We have also shown that this simple technique, which does not require any training data, produces annotations at approximately the same cost of a metadata-trained, supervised cascading machine classifier, or about 70% cheaper than 5-vote majority-vote aggregation. We expect that future work will combine this technique for seed data creation with algorithms such as Active Learning to create corpora large enough for machine learning, at a reduced cost.

Acknowledgments

This work has been supported by the Volkswagen Foundation as part of the Lichtenberg-Professorship Program under grant No. I/82806, and by the Center for Advanced Security Research (`www.cased.de`).

References

Daniel Bär, Torsten Zesch, and Iryna Gurevych. 2011. A reflective view on text similarity. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, pages 515–520, Hissar, Bulgaria.

Gustavo E.A.P.A. Batista, Ronaldo C. Prati, and Maria Carolina Monard. 2004. A study of the behavior of several methods for balancing machine learning training data. *ACM Sigkdd Explorations Newsletter*, 6(1):20–29.

Lubomir Bourdev and Jonathan Brandt. 2005. Robust object detection via soft cascade. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 236–243, Washington D.C., USA.

Sabine Buchholz and Javier Latorre. 2011. Crowdsourcing preference tests, and how to detect cheating. In *Proceedings of the 12thAnnual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 3053–3056, Florence, Italy.

Krista Casler, Lydia Bickel, and Elizabeth Hackett. 2013. Separate but equal? A comparison of participants and data gathered via Amazons MTurk, social media, and face-to-face behavioral testing. *Computers in Human Behavior*, 29(6):2156–2160.

Barnan Das, Narayanan C. Krishnan, and Diane J. Cook. 2014. Handling imbalanced and overlapping classes in smart environments prompting dataset. In Katsutoshi Yada, editor, *Data Mining for Service*, pages 199–219. Springer, Berlin Heidelberg.

A. P. Dawid and A. M. Skene. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics*, 28(1):20–28.

Dmitriy Dligach and Martha Palmer. 2011. Reducing the need for double annotation. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 65–73, Stroudsburg, Pennsylvania.

Julie S. Downs, Mandy B. Holbrook, Steve Sheng, and Lorrie Faith Cranor. 2010. Are your participants gaming the system?: Screening Mechanical Turk workers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2399–2402, Atlanta, Georgia.

Charles Elkan. 2001. The foundations of cost-sensitive learning. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pages 973–978, San Francisco, California.

Pei-Yun Hsueh, Prem Melville, and Vikas Sindhwani. 2009. Data quality from crowdsourcing: a study of annotation selection criteria. In *Proceedings of the NAACL HLT 2009 workshop on active learning for natural language processing*, pages 27–35, Boulder, Colorado.

Panagiotis G. Ipeirotis, Foster Provost, and Jing Wang. 2010. Quality management on Amazon Mechanical Turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 64–67, Washington D.C., USA.

Emily K. Jamison and Iryna Gurevych. 2013. Headerless, quoteless, but not hopeless? using pairwise email classification to disentangle email threads. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 327–335, Hissar, Bulgaria.

Hyun Joon Jung and Matthew Lease. 2012. Improving quality of crowdsourced labels via probabilistic matrix factorization. In *Proceedings of the 4th Human Computation Workshop (HCOMP) at AAAI*, pages 101–106, Toronto, Canada.

Abhimanu Kumar and Matthew Lease. 2011. 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, Hong Kong, China.

George K. Mikros and Eleni K. Argiri. 2007. Investigating topic influence in authorship attribution. In *Proceedings of the SIGIR 2007 International Workshop on Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection, PAN 2007*, Amsterdam, Netherlands. Online proceedings.

Stefanie Nowak and Stefan Rüger. 2010. How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation. In *Proceedings of the international conference on Multimedia information retrieval*, pages 557–566, Philadelphia, Pennsylvania.

David Oleson, Alexander Sorokin, Greg P. Laughlin, Vaughn Hester, John Le, and Lukas Biewald. 2011. Programmatic gold: Targeted and scalable quality assurance in crowdsourcing. *Human computation*, 11:11.

Vikas C. Raykar, Balaji Krishnapuram, and Shipeng Yu. 2010. Designing efficient cascaded classifiers: tradeoff between accuracy and cost. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 853–860, New York, NY.

Victor S. Sheng, Foster Provost, and Panagiotis G. Ipeirotis. 2008. Get another label? Improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 614–622, Las Vegas, Nevada.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. 2008. Cheap and fast—but is it good?: Evaluating non-expert annotations for natural language tasks. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*, pages 254–263, Honolulu, Hawaii.

Alexander Sorokin and David Forsyth. 2008. Utility data annotation with Amazon Mechanical Turk. *Urbana*, 51(61):820.

Jon Sprouse. 2011. A validation of Amazon Mechanical Turk for the collection of acceptability judgments in linguistic theory. *Behavior research methods*, 43(1):155–167.

Alexey Tarasov, Sarah Jane Delany, and Brian Mac Namee. 2014. Dynamic estimation of worker reliability in crowdsourcing for regression tasks: Making it work. *Expert Systems with Applications*, 41(14):6190–6210.

Paul A. Viola and Michael J. Jones. 2001. Rapid object detection using a boosted cascade of simple features. In *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 511–518, Kauai, Hawaii.

Peter Welinder and Pietro Perona. 2010. Online crowdsourcing: rating annotators and obtaining cost-effective labels. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 25–32, San Francisco, California.

Omar F. Zaidan and Chris Callison-Burch. 2011. Crowdsourcing translation: Professional quality from non-professionals. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1220–1229, Portland, Oregon.