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 original 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-
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 crowdsourced 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 crowdsourced 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 crowdsourced 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 crowdsourced 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 crowdsourcing 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 Bår 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:

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

and number of common-class instances:

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

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 Mechanical Turk (MTurk)$^3$ annotations for each of 100 pairs of emails, at a cost of US$0.033$ per annotation. Standard quality measures employed to reduce spam annotations included over 2000 HITs (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:

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

$^3$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 so-
licited five MTurk annotations for each of 750
turn/edit text pairs at a cost of US$0.044 per anno-
tation. Measures for Wikipedia turn/edit pairs in-
cluded 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 qualifica-
tion test of sample pairs. The cost of a rare-class
instance is:
\[
\frac{\text{US$0.044} \times 5 \text{ annotators}}{0.01 \text{ freq}} = \text{US$22}
\]

SentPairs Dataset Cost  The SentPairs datset con-
sists 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 esti-
mated a cost of US$0.01 per annotation. The an-
tnotations came from Crowdflower\(^7\). Bär et al.
(2011) used a number of quality assurance mech-
anisms, such as worker reliability and annota-
tion correlation. The cost of a rare-class in-
stance varied between classes, due to class fre-
cuency variation, from instance\(_{\text{class2}}\) = US$0.027 to
instance\(_{\text{class5}}\) = 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 confirm-
ing 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 ex-
periment results:
TP is the number of true positives (rare-class) dis-
covered. The fewer TP’s discovered, the less likely
the resulting corpus will represent the original data
in an undistorted manner.
\(P_{\text{rare}}\) is the precision over rare instances: \(\frac{TP}{TP + FP}\)
Lower precision means lower confidence in the pro-
duced dataset, because the “rare” instances we found
might have been misclassified.

\(\text{AvgA}\) is the average number of annotations needed
for the system to label an instance common-class.

The normalized cost is the estimated cost of acquir-
ing a rare instance:
\[
\text{AvgA} \times \text{annoCost} \times \text{Recall}_{\text{rare}}
\]

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 sec-
tion, 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 clas-
sified as potentially rare received another annotation
and continued through the next cascade, while in-
stances classified as common were discarded. Dis-
carding instances before the end of the cascade can
reduce the total number of needed annotations, and
therefore lower the total cost. This cascade mod-
els the observation (see Section 5) that it is cheap to
confirm suspected rare-class instances, but it is ex-
pensive to weed out common-class instances.

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

6.1 Instances
Each experimental instance consisted of features de-
duced from the metadata of one or more crowd-
sourced 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 an-
notation. 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
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.

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
the experiment.

class annotations. Class i for each instance has more annotations. A gold standard rare instances come greater than 2, we assume the instance is no.

cascade except that a single rule replaces each clas-
cascade functions similarly to the trained classifier of our previous trained classifier. The rule-based experiments, we evaluate a rule-based cascade in place of our previous trained classifier. 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 a1, a2, ..., ak and the probability (Pr) of five non-common-class annotations. Class c is the common class. We always need a first annotation: Pr(a1, ≠ c) = 1. 

\[
\text{AvgA}_i = \sum_{j=1}^{k} \prod_{k=1}^{j} \Pr(a_k, \neq c)
\]

We define Precisionrare (Prar) as the probability that instance i with 5 common annotations a1, a2, ..., ak is not a rare-class instance:

\[
P_{\text{rare}, i} = \Pr(TP|(a_1...a_k = \text{rare})) = 1 - \Pr(FP|(a_1...a_k = \text{rare}))
\]

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 \(P_{\text{rare}} = \frac{TP}{TP+FP}\).

Table 4: SentPairs results on the trained cascade.

<table>
<thead>
<tr>
<th>Features</th>
<th>TPs</th>
<th>Prar</th>
<th>AvgA</th>
<th>Norm cost</th>
<th>Savings(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>17</td>
<td>1.00</td>
<td>0.49</td>
<td>$0.64</td>
<td></td>
</tr>
<tr>
<td>anno</td>
<td>14</td>
<td>0.72</td>
<td>2.4545</td>
<td>$0.15</td>
<td>66±7</td>
</tr>
<tr>
<td>worker</td>
<td>14</td>
<td>0.63</td>
<td>2.7937</td>
<td>$0.16</td>
<td>64±8</td>
</tr>
<tr>
<td>dur</td>
<td>10</td>
<td>0.52</td>
<td>2.7111</td>
<td>$0.18</td>
<td>58±11</td>
</tr>
<tr>
<td>worker &amp; anno</td>
<td>15</td>
<td>0.82</td>
<td>2.3478</td>
<td>$0.12</td>
<td>73±8</td>
</tr>
<tr>
<td>worker &amp; dur</td>
<td>6</td>
<td>0.4</td>
<td>2.7578</td>
<td>$0.38</td>
<td>14±23</td>
</tr>
<tr>
<td>dur &amp; anno</td>
<td>16</td>
<td>0.72</td>
<td>2.4887</td>
<td>$0.14</td>
<td>69±10</td>
</tr>
<tr>
<td>all</td>
<td>17</td>
<td>0.82</td>
<td>2.4408</td>
<td>$0.12</td>
<td>73±5</td>
</tr>
</tbody>
</table>

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 a1, a2, ..., ak and the probability (Pr) of five non-common-class annotations. Class c is the common class. We always need a first annotation: Pr(a1, ≠ c) = 1.

\[
\text{AvgA}_i = \sum_{j=1}^{k} \prod_{k=1}^{j} \Pr(a_k, \neq c)
\]

We define Precisionrare (Prar) as the probability that instance i with 5 common annotations a1, a2, ..., ak is not a rare-class instance:

\[
P_{\text{rare}, i} = \Pr(TP|(a_1...a_k = \text{rare})) = 1 - \Pr(FP|(a_1...a_k = \text{rare}))
\]

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 \(P_{\text{rare}} = \frac{TP}{TP+FP}\).

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

<table>
<thead>
<tr>
<th>Class = N if:</th>
<th>TP</th>
<th>P_{rare}</th>
<th>AvgA</th>
<th>NormCost</th>
<th>Savings(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>128</td>
<td>1.00</td>
<td></td>
<td>$22.0</td>
<td></td>
</tr>
<tr>
<td>no &gt; 0</td>
<td>39</td>
<td>0.95</td>
<td>1.61</td>
<td>$7.09</td>
<td>68±16</td>
</tr>
<tr>
<td>no &gt; 1</td>
<td>39</td>
<td>0.85</td>
<td>2.86</td>
<td>$12.6</td>
<td>43±19</td>
</tr>
<tr>
<td>no &gt; 2</td>
<td>39</td>
<td>0.73</td>
<td>3.81</td>
<td>$17.65</td>
<td>24±15</td>
</tr>
<tr>
<td>(nor+c)=0</td>
<td>22</td>
<td>0.98</td>
<td>1.35</td>
<td>$10.56</td>
<td>52±20</td>
</tr>
<tr>
<td>(nor+c)=1</td>
<td>33</td>
<td>0.93</td>
<td>2.55</td>
<td>$13.25</td>
<td>40±18</td>
</tr>
<tr>
<td>(nor+c)=2</td>
<td>35</td>
<td>0.85</td>
<td>3.56</td>
<td>$17.44</td>
<td>21±15</td>
</tr>
</tbody>
</table>

Table 6: Wiki results: no ambiguous instances.

Table 7: Email results: rule-based cascade.

<table>
<thead>
<tr>
<th>Class = N if:</th>
<th>TP</th>
<th>P_{rare}</th>
<th>AvgA</th>
<th>NormCost</th>
<th>Savings(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>34</td>
<td>1.00</td>
<td></td>
<td>$16.5</td>
<td></td>
</tr>
<tr>
<td>no &gt; 0</td>
<td>32</td>
<td>1.00</td>
<td>1.07</td>
<td>$3.52</td>
<td>79±4</td>
</tr>
<tr>
<td>no &gt; 1</td>
<td>32</td>
<td>0.99</td>
<td>2.11</td>
<td>$6.95</td>
<td>58±7</td>
</tr>
<tr>
<td>no &gt; 2</td>
<td>32</td>
<td>0.98</td>
<td>3.12</td>
<td>$10.31</td>
<td>38±6</td>
</tr>
<tr>
<td>(nor+c)=0</td>
<td>30</td>
<td>1.00</td>
<td>1.04</td>
<td>$3.67</td>
<td>78±5</td>
</tr>
<tr>
<td>(nor+c)=1</td>
<td>32</td>
<td>0.99</td>
<td>2.07</td>
<td>$6.83</td>
<td>59±6</td>
</tr>
<tr>
<td>(nor+c)=2</td>
<td>32</td>
<td>0.99</td>
<td>3.08</td>
<td>$10.16</td>
<td>38±5</td>
</tr>
</tbody>
</table>

Table 8: SentPairs results: rule-based cascade.

<table>
<thead>
<tr>
<th>Class = N if:</th>
<th>TP</th>
<th>P_{rare}</th>
<th>AvgA</th>
<th>NormCost</th>
<th>Savings(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>16</td>
<td>1.00</td>
<td></td>
<td>$0.47</td>
<td></td>
</tr>
<tr>
<td>no &gt; 0</td>
<td>16</td>
<td>0.99</td>
<td>1.98</td>
<td>$0.09</td>
<td>80±9</td>
</tr>
<tr>
<td>no &gt; 1</td>
<td>16</td>
<td>0.96</td>
<td>3.83</td>
<td>$0.18</td>
<td>62±15</td>
</tr>
<tr>
<td>no &gt; 2</td>
<td>16</td>
<td>0.9</td>
<td>5.47</td>
<td>$0.26</td>
<td>45±17</td>
</tr>
<tr>
<td>(nor+c)=0</td>
<td>0</td>
<td>1.0</td>
<td>1.23</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(nor+c)=1</td>
<td>0</td>
<td>1.0</td>
<td>2.45</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(nor+c)=2</td>
<td>1</td>
<td>0.99</td>
<td>3.65</td>
<td>$2.74</td>
<td>-484±162</td>
</tr>
</tbody>
</table>

Table 9: SentPairs 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 crowdsourcer 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 crowdsourcing 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

<table>
<thead>
<tr>
<th></th>
<th>AvgA</th>
<th>NormCost</th>
<th>Savings(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>0.99</td>
<td>1.96</td>
<td>$0.09</td>
</tr>
<tr>
<td>no &gt; 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>no &gt; 1</td>
<td>0.95</td>
<td>3.77</td>
<td>$0.17</td>
</tr>
<tr>
<td>no &gt; 2</td>
<td>0.89</td>
<td>5.37</td>
<td>$0.24</td>
</tr>
<tr>
<td>(nor+ct) &gt; 0</td>
<td>2.00</td>
<td>1.27</td>
<td>$0.48</td>
</tr>
<tr>
<td>(nor+ct) &gt; 1</td>
<td>1.00</td>
<td>2.54</td>
<td>$0.19</td>
</tr>
<tr>
<td>(nor+ct) &gt; 2</td>
<td>1.30</td>
<td>3.8</td>
<td>$0.22</td>
</tr>
</tbody>
</table>

Table 11: SentPairs results: rule-based cascade.

<table>
<thead>
<tr>
<th>Ambiguous instances</th>
<th>Unambiguous instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anno, y et n # inst</td>
<td>Anno, y et n # inst</td>
</tr>
<tr>
<td>3 0 2</td>
<td>35  5 0 0 22</td>
</tr>
<tr>
<td>3 1 1</td>
<td>30  4 1 0 11</td>
</tr>
<tr>
<td>2 2 1</td>
<td>19  4 0 1 28</td>
</tr>
<tr>
<td>2 1 2</td>
<td>39  3 2 0 2</td>
</tr>
</tbody>
</table>

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 ac-
cordingly, 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 com-
plaining 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 an-
notation 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 re-
duction 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 approxi-
ately the same cost of a metadata-trained, super-
vised 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 Cen-
ter for Advanced Security Research (www.cased.
d).

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 behav-
ior of several methods for balancing machine learning
training data. ACM Sigkdd Explorations Newsletter,

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

Sabine Buchholz and Javier Latorre. 2011. Crowdsourc-
ing preference tests, and how to detect cheating. In
Proceedings of the 12th Annual Conference of the In-
ternational Speech Communication Association (IN-
TERSPEECH), 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

Barnan Das, Narayanan C. Krishnan, and Diane J. Cook.
2014. Handling imbalanced and overlapping classes in
smart environments prompting dataset. In Katsutoshi
Springer, Berlin Heidelberg.

A. P. Dawid and A. M. Skene. 1979. Maximum likeli-
hood estimation of observer error-rates using the EM


