Predicting worker disagreement for more effective crowd labeling

Stefan Räbiger*, Gizem Gezici*, Yücel Saygın*, and Myra Spiliopoulou†
*Faculty of Engineering and Natural Sciences
Sabancı University, Istanbul (Turkey)
Email: {stefan, gizemgezici, ysaygin}@sabanciuniv.edu
†Knowledge Management and Discovery Lab
Otto-von-Guericke University, Magdeburg (Germany)
Email: myra@ovgu.de

***Abstract*—Crowdsourcing is a popular mechanism used for labeling tasks to produce large corpora for training. However, producing a reliable crowd labeled training corpus is challenging and resource consuming. Research on crowdsourcing has shown that label quality is much affected by worker engagement and expertise. In this study, we postulate that label quality can also be affected by inherent ambiguity of the documents to be labeled. Such ambiguities are not known in advance, of course, but, once encountered by the workers, they lead to disagreement in the labeling – a disagreement that cannot be resolved by employing more workers. To deal with this problem, we propose a crowd labeling framework: we train a disagreement predictor on a small seed of documents, and then use this predictor to decide which documents of the complete corpus should be labeled and which should be checked for document-inherent ambiguities before assigning (and potentially wasting) worker effort on them. We report on the findings of the experiments we conducted on crowdsourcing a Twitter corpus for sentiment classification.**

***Index Terms*—worker disagreement, crowdsourcing, dataset quality, label reliability, tweet ambiguity**

I. Introduction

Crowdsourcing is a popular mechanism to obtain large-scale labeled corpora for supervised learning techniques. Hence, it is crucial that crowd workers are reliable and provide accurate labels. To that end, multiple reliability indicators like the annotation behavior over time [1] or consistency [2], have been proposed for workers. Consistency might be affected by training, expertise, or fatigue emerging during a crowdsourcing task. In [3], the authors report that workers produce more reliable labels if they must explain their rationale for choosing a specific label before assigning it. Psychological effects such as the Dunning-Kruger effect [4] (crowd workers might overestimate their expertise w.r.t. a topic and therefore try to compensate for it with general knowledge), also affect the reliability of workers. These studies among others assume that the key factors of success in crowdsourcing are properties of the workers - either intrinsic ones like experience, or extrinsic ones like adequate training (having positive influence) or fatigue (negative influence). While we agree with these and the importance of a clear task specification [5], we postulate that the success of a crowdsourcing task also depends on properties of the documents to be labeled by the workers. Consider for example the typical crowdsourcing scenario of deciding whether a short text document like a tweet has positive or negative sentiment, and assume that a worker encounters the following tweet:

```
Quoting Michelle. More points! "Go low.
Shawty, I go high" while I bring up
your racist past. #debatenight
```

Evidently, this tweet is rather difficult to label, so it might be fair to have the experimenter look at it and decide whether it should be indeed labeled or not. Obviously, inspecting all documents in advance is impractical, hence the goal of our proposed method is to identify those documents to be inspected because they are expected to provoke high disagreement (and thus waste worker budget) if labeled.

Our contribution is a new crowdsourcing methodology that a) improves the reliability of crowdsourced corpora and b) enhances the predictor performance that is learned on those corpora. Our method trains a disagreement predictor on a small seed set that separates among different levels of disagreement, learning on the properties of the documents, rather than the properties of the workers. The size of the seed set is then iteratively increased based on the disagreement predictor. The predictor then estimates the level of disagreement in each unlabeled document of the corpus and all documents with worker disagreement are considered ambiguous and it is left to the experimenter how to deal with them, e.g. by removing them or letting experts label them. Only those documents with no disagreement will be crowdsourced.

Unlike existing studies that have investigated the link between document difficulty and label reliability in crowdsourcing [6], our method is applied as a preprocessing step before crowdsourcing the remaining documents. Hence both methods complement each other. Upon combination, the prior for document difficulty in the method proposed by Whitehall et al. could be adjusted toward easy (=non-ambiguous) documents due to our method being applied as a preprocessing step. Our approach aligns with the methods that investigate the issue of *aleatoric uncertainty* as opposed to *epistemic uncertainty*: as

978-1-5386-5090-5/18/$31.00 ©2018 IEEE

the authors of [7] point out, epistemic uncertainty on a given outcome (here: the document's label) can be reduced by acquiring additional expert opinions, while aleatoric uncertainty cannot be reduced, because the additional experts will have also diverging opinions on the label. Thus, our method allows that documents with disagreement are not given to the workers.

Our results using a sentiment analysis task on Twitter suggest that removing tweets with disagreement improves the sentiment predictor's performance, while acquiring more labels for tweets with disagreement does not.

II. Related work

Producing high-quality labels at moderate costs is the main advantage of crowdsourcing. The link between a multitude of different traits of crowd workers, also known as human factors, w.r.t. label quality and reliability has been investigated in the past. They include, but are not limited to, examining the influence of framing, i.e. sharing the purpose of the labeling task with crowd workers [8], how worker expertise affects label reliability [9], how the reliability of labels that workers assign develops over time [10], and also the reliability of crowd workers. For the latter problem, characteristic patterns of temporal behavior of low-quality workers have been identified which may be utilized to remove such contributions [11].

Our work is based on the assumption that "[crowd worker] disagreement is not noise, but signal" [12] because we use it as an indicator of difficult documents. Worker disagreement in crowdsourcing is investigated in different contexts. For word sense annotations it was found that it is easier to predict high disagreement than lower levels of disagreement [13], which is why we model it as a binary classification task. Generalizability theory is employed to analyze different factors (called "facets") of an annotation experiment to identify those factors that contribute most to high worker disagreement [14]. Others find that training workers reduces disagreement [15] and that some strategies for training workers are more promising [16]. It was shown that high/low Kappa/Krippendorf's alpha values, which both measure worker disagreement, do not necessarily correlate with predictor performance [17]. For example, low worker disagreement could have been artificially achieved by workers preferring one specific label over others. Predictors trained on these data would also be biased and therefore perform poorly on unknown data. Hence, training workers comes with its own risks: providing biased examples to workers might introduce biased labels, s.t. one label is preferred over others. Since we are using a subjective sentiment analysis task in this study, we do not provide sample documents from the dataset to explain the labels, just a short, general description with imaginary, simple documents to avoid introducing any bias.

Closer to our study are works that investigate how task difficulty affects the crowdsourcing task. In [18] the authors seek ways to incentivize crowd working for labeling tasks of varying difficulty. To obtain more reliable corpora, in [19] an algorithm is proposed which allocates more budget to difficult (sarcastic) tweets so that more crowd workers can label those. They infer tweet difficulty from worker disagreement.

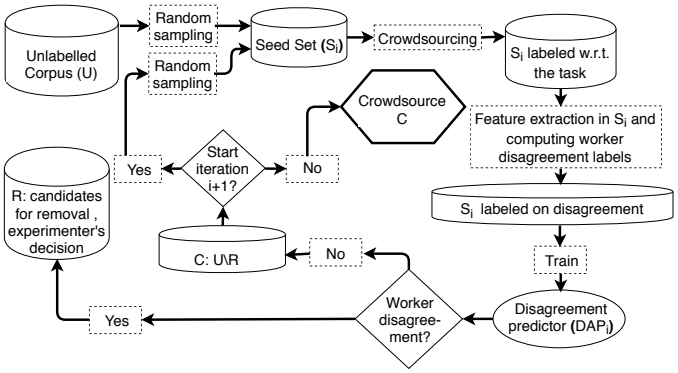

Fig. 1. Schematic overview of our proposed methodology to obtain a more reliable corpus C for crowdsourcing, where i refers to the i^{th} iteration as described in the text.

However, their objective is to find the tweets that must be labeled by more people while our objective is to find the tweets that may be treated differently before being given out for crowdsourcing at all. Therefore, we are the first to demonstrate how predictor performance is affected by removing tweets with high disagreement compared to allotting more workers to them.

III. Methodology

We propose a multi-stage iterative methodology, which is depicted in Fig. 1. Given an unlabeled corpus U, we start with a small, randomly sampled seed set (see top part of Fig. 1) to be labeled by the crowd workers w.r.t. a certain labeling task, e.g. sentiment analysis (see top-right corner of Fig. 1). For each document in the seed set, we count the labels assigned to it by the workers and assess whether there is disagreement in the workers' decisions. We thus turn the seed set into a training set on worker disagreement (see right part of Fig. 1). Then, we train a disagreement predictor (see bottom-right corner of Fig. 1) which estimates the worker disagreement in the unlabeled documents. Documents on which workers are expected to agree are moved to corpus C. Otherwise they are moved to corpus R and it is the experimenter's choice how to proceed with them, e.g. removing them, letting experts label them, labeling every n^{th} document, etc. The experimenter may also decide for a further iteration with an expanded seed set (see middle part of Fig. 1), thus refining the disagreement predictor. After all iterations are completed, only documents remaining in corpus C will be labeled by crowd workers. In the following subsections, we describe the details of our approach.

A. Modeling disagreement among crowd workers

A worker assigning a label to a document is called a *vote*. If there are n votes for a document, n different workers labeled it. Since the true label of a document might be unknown, we use the majority label according to the majority voting scheme instead. We employ two levels of disagreement in this study: disagreement or no disagreement.

Definition 1. *Provided that there are n votes available for a document, there is disagreement if the majority label received not more than 50% of the votes. Otherwise there is no disagreement.*

This definition depends only on the number of workers who labeled a document, but not on the number of classes that exist. For example, if a document received eight votes, i.e. eight workers labeled it, we conclude that the workers disagree on its label if the majority label was assigned four or less times. This is independent of the number of classes in the labeling task. Based on the above definition we consider documents with disagreement as *ambiguous* and others as *unambiguous*.

B. Disagreement predictor

The disagreement predictor DAP_i plays an important role in our method as it reduces the size of the corpus to be labeled by the crowd. The initial seed set S_0 is created from the unlabeled corpus U by randomly selecting a set of n documents, N_0 (line 8 in Algorithm 1), which are then labeled by crowd workers. Algorithm 2 then derives the disagreement labels according to Definition 1 turning N_0 into S_0. DAP_0 is trained on S_0 before predicting the disagreement labels for all unlabeled documents $U \setminus S_0$. These documents are then either moved to corpus R (disagreement) or corpus C (no disagreement) (line 14-17 in Algorithm 1). Therefore, C contains only the tweets $U \setminus (R \cup S_0)$. If the experimenter prefers to increase the performance of DAP_0 (line 21), another iteration begins, but this time documents are randomly sampled from C instead of U (line 19). The stopping criterion is discussed separately in the next section. In the next iteration, S_1 is created by sampling another n documents from C, N_1. After crowdsourcing and deriving the disagreement labels, N_1 is merged with S_0 resulting in S_1. In general, we obtain S_i in the i^{th} iteration as $S_i = N_i \cup S_{i-1}$. DAP_i is then trained on S_i and it predicts the disagreement of the remaining tweets in C to further reduce the size of C. After all iterations only the documents remaining in C will be crowdsourced. The *ambiguous* documents in corpus R allow experimenters to decide on a case-by-case basis if it is beneficial to let experts label those documents, label only every n^{th} document, completely remove them etc. We evaluate the initial effectiveness of DAP_0 according to research question RQ_0^{1a} (see Table II) to test how well disagreement may be predicted.

C. Stopping criterion for expanding the seed set

It might be necessary to expand S_i iteratively (line 6 in Algorithm 1) to improve the performance of DAP_i, e.g. due to high class imbalance or feedback from crowd workers who identified flaws in the task design. One simple option to stop the expansion would be the experimenter's budget constraints: crowd labeling N_i consumes a certain amount of the budget in each iteration i, thus an experimenter could know in advance when to stop expanding S_i. Another possible stopping criterion for practical use would be monitoring corpus R, which stores removed documents, and checking after each iteration if the number of documents with predicted disagreement has decreased. This information might suffice for experimenters to decide about continuing with the expansion or not. We implicitly assume that training DAP_i on the expanded S_i yields better performance as more training data becomes available. Since our method relies on this assumption, we test it in research question RQ_0^2 (see Table II).

Algorithm 1 Iteratively Estimating the Level of Disagreement to Remove Ambiguous Documents.

```
1:  Input: Corpus of unlabeled documents (U).
2:  Output: Set of documents to be labeled via crowdsourcing
    (C), set of ambiguous documents (R)
3:  S ← Ø                                  ▷ seed set of previous iteration
4:  R ← Ø
5:  iteration i = 0;
6:  repeat
7:      C ← Ø
8:      N_i ← randSample(U \ S, n)         ▷ pick n documents
9:      crowdsource(N_i)
10:     S_i ← createTrainingSet(N_i, S)    ▷ see Algorithm 2
11:     DAP_i.train(S_i)                   ▷ train on disagreement labels
12:     for each document d in U \ S_i do
13:         label ← DAP_i.predict(d)
14:         if label == 'yes' then
15:             R ← R ∪ d
16:         else
17:             C ← C ∪ d
18:     S ← S_i
19:     U ← C                              ▷ label propagation
20:     i = i + 1
21: until experimenter stops               ▷ see section about the
                                             stopping criterion
22: return C, R
```

Algorithm 2 Creation of S for the disagreement predictor.

```
1:  Input: Set of documents with crowdsourced labels (N),
    seed set with one disagreement label per document (S)
2:  Output: Set of documents with one disagreement label
    each.
3:  function createTrainingSet(N, S)
4:      for each document d in N do
5:          n ← allVotes(d)                ▷ total votes
6:          m ← majVotes(d)                ▷ #votes for majority label
7:          label ← 'no'                   ▷ no disagreement
8:          if m ≤ n/2 then
9:              label ← 'yes'              ▷ disagreement
10:         d.setDisagreement(label)
11:     return N ∪ S
```

IV. Evaluation framework

This section describes how we created a crowdsourced corpus for a hierarchical sentiment analysis task on Twitter. Additionally, we describe the features used in the disagreement predictor and the sentiment predictor. Both are necessary for

evaluating our approach. Since sentiment analysis is subjective and tweets are short, ambiguity is likely to occur, which makes it a suitable task for testing our methodology. Formulating the task as a hierarchical one allows us to focus on the sentiment of relevant tweets only. Specifically, workers assigned as sentiment labels for relevant tweets either *positive*, *negative*, or *neutral*. Irrelevant tweets are given the label *irrelevant*.

A. Corpus collection

We use as seed set S_0 the dataset collected in [1] containing tweets that were posted during the first debate between Hillary Clinton and Donald Trump in the US presidential election campaign 2016. The dataset encompasses 500 tweets labeled hierarchically in terms of sentiment. With the provided tweet IDs from [1] we downloaded the respective metadata using the Twitter API and we collected another 19.5k tweets that were posted during the first debate between Hillary Clinton and Donald Trump. Following the preprocessing protocol from [1], those 19.5k tweets neither contained URLs nor attachments like pictures. This way, sentiment can only be expressed directly in the texts instead of conveying it through linked websites or attached videos/pictures. To illustrate how these tweets look like, we present two tweets. The crowd workers agreed on the sentiment of the first one:

```
Please tell me we have other options
for president. These 2 are fruit loops!
\#DebateNight \#Doomed \#VoteForPedro
```

But they disagreed on the sentiment of the second one below:

```
I can't take either seriously until
Lester Holt asks the real question
in this debate: is a hot dog a
sandwich? \#debatenight \#teachthetruth
```

B. Labeling the seed set

Since the hierarchical labeling scheme is important to understand how we derive worker disagreement, we briefly explain the scheme utilized in [1]. There are in total three levels in the hierarchy. On the first level, workers choose between the labels *relevant* and *irrelevant* to indicate a tweet's relevance regarding the US presidential debate. Afterwards workers are prompted to select either *factual* (which corresponds to *neutral*) or *non-factual* on the second level. If the latter label is chosen, workers are presented the final set of labels, *positive* and *negative* on the third hierarchy level. If workers chose *irrelevant* on the first level, all labels assigned on the second and third level were discarded. Each one of the 500 tweets received between 4-30 votes.

C. Building crowdsourced corpora

For determining the worker disagreement in S_0 for tweet t, we devised the following scoring function yielding values between 0 (no agreement) and 1 (perfect agreement) using majority voting to obtain ground truth labels:

$$a(t) = \sum_{i \in\ Levels} \frac{|workers_{maj}|}{|workers_i|} * \frac{|workers_{maj}|}{total_{maj}}. \quad (1)$$

where $workers_{maj}$ are the crowd workers who assigned the majority label on hierarchy level i, $workers_i$ are the workers who labeled t on level i, $total_{maj}$ is the total number of workers across all hierarchy levels that assigned majority labels, and $Levels$ is the set of hierarchy levels in the labeling scheme, in our case $Levels = \{1, 2, 3\}$. The first term in Equation 1 describes the fraction of workers who agreed on the majority label at level i, while the second expression accounts for the overall contribution of level i to the agreement score. Whenever there is a tie between majority labels at level i, $total_{maj}$ is incremented by one. This reduces the contribution of hierarchy levels, that have no ties, to the overall agreement score, which generally leads to lower scores for tweets with ties. A small example illustrates how Equation 1 works: suppose that four workers labeled tweet $t1$ and assigned the labels:

- First hierarchy level: *relevant*, *relevant*, *relevant*, *relevant*
- Second hierarchy level: *factual*, *non-factual*, *non-factual*, *non-factual*
- Third hierarchy level: -, *negative*, *negative*, *positive*

The label "-" indicates that no label has to be assigned on this hierarchy level because the tweet is already *factual*, i.e. neutral. In total, nine workers assigned the majority labels (four on the first level, three on the second level, two on the third level), so $total_{maj} = 9$. The majority labels for $t1$ are *relevant*, *non-factual*, and *negative*, leading to $a(t1) = 4/4 * 4/9 + 3/4 * 3/9 + 2/2 * 2/9 = 0.92$. After computing $a(t)$, computing the disagreement score for tweet t becomes: $1 - a(t)$. We then bin the computed disagreement scores to three disagreement levels: *low*, *medium*, and *high* and train DAP_0 on S_0 with those derived labels.

In the next step, DAP_0 predicted the worker disagreement in the remaining 19.5k tweets. To test the performance of DAP_0, we created three corpora - LOW, MEDIUM, and HIGH. LOW (MEDIUM) (HIGH) contains 1k randomly selected tweets with predicted disagreement *low* (*medium*) (*high*). To evaluate how well DAP_0 performs, we request labels from AMT for all three corpora where each tweet in HIGH is labeled by eight different workers, whereas tweets from MEDIUM and LOW are labeled by four workers each. We allocate more budget to HIGH since it is the most promising corpus to contain tweets with disagreement, which we want to analyze. Building these three corpora allows us to analyze DAP_0's performance on real data in research question RQ_0^{1b} (see TableII). To ensure the quality of the crowd workers, we only permitted workers with an acceptance rate of at least 90% to participate. They were also provided with instructions on the labeling task and an imaginary sample tweet per class label. Before acceptance we inspected submitted micro-tasks manually.

We note that we initially chose the worker disagreement labels for S_0 as *low*, *medium*, and *high*. For our crowdsourcing experiment we converted the hierarchical labeling scheme from [1] into a more suitable flat one using the labels *positive*, *negative*, *neutral* for *relevant* tweets, and *irrelevant* otherwise.

At this time we also changed worker disagreement from three to two levels because we are only interested in tweets with and without disagreement. These two corrections allowed using the more intuitive majority voting scheme (see Definition 1) because (1) does not yield continuous scores for a flat labeling scheme. In other words, (1) was only used for creating the three corpora, but otherwise the flat scheme and binary worker disagreement labels were used throughout the paper. The flat scheme was also applied to S_0 after the three corpora were created.

D. Features for disagreement and sentiment classification

Table I shows the features that are used by the sentiment predictor STP and the disagreement predictor DAP_i. We note that due to hyperparameter optimization not necessarily all features are utilized by each predictor. Since we are only interested in sentiment w.r.t. a specific topic (presidential debate), we exploit the similarity between a query and tweets to determine a tweet's relevance. The query is the same for all tweets and we set it to "donald trump hillary clinton political election discussion campaign" in this study.

As shown in Table I, we exploit tweet sentiment and compute polarity values from the given text by using four different resources: two online tools, namely Watson[3] and TextBlob[4], and two lexicons, SentiWordNet (SWN) [24] which is a domain-independent lexicon and the SemEval-2015 English Twitter Lexicon (TWL) [25] which is specifically tailored to Twitter. In terms of sentiment, we also utilize subjective word lists proposed by [26]. Please note that we computed features $F_2 - F_{42}$ for the whole tweet as well as for the first and second half separately. Otherwise 13 features instead of 39 would have sufficed for our representation. Regarding the syntactic features, we obtain POS tags from Rosette[5] and NERs from Rosette and Watson.

Since there is a correlation between sarcastic tweets and worker disagreement [19], we include sarcasm-related features ($F_{59} - F_{67}$) as sarcasm increases ambiguity. On top of these, we generate ten topics from the whole corpus by using LDA [27], since topic features may also convey sarcasm-related information. Finally, we include word embeddings, specifically pre-trained Glove vectors [23] for Twitter[6], which may preserve semantic information.

Evaluating STP allows to investigate our core claim with research questions RQ_0^3 and RQ_0^4 (see Table II), namely that documents (here: tweets) affect predictor performance negatively and removing them might be helpful.

[1]http://saifmohammad.com/WebPages/SCL.html#ETSL

[2]https://nlp.stanford.edu/IR-book/html/htmledition/query-term-proximity-1.html

[3]https://www.ibm.com/watson/developercloud/natural-language-understanding/api/v1

[4]https://textblob.readthedocs.io/en/dev

[5]https://developer.rosette.com/api-guide

[6]https://nlp.stanford.edu/projects/glove/

TABLE I
OVERVIEW OF FEATURES USED FOR SENTIMENT AND DISAGREEMENT PREDICTORS.

Group Name	Feature	Description
Polarity	F_1	Watson Sentiment
	F_2-F_7	Avg. pol. and ratio (TextBlob)
	F_8-F_{21}	Min/Max/Avg/Dominant pol. and ratio (SWN)
	F_{22}-F_{33}	Min/Max/Avg pol. & ratio (TWL[1])
Subjective Words	F_{31}-F_{42}	#Pos./Neg. words and their ratio
TF*IDF	F_{43}-F_{47}	Sum/Mean/Min/Max variance of TF*IDF scores of words
Syntactic	F_{48}-F_{55}	#POS tags (nn, jj, rb, vb) and ratio
	F_{56}	#NERs
	F_{57}	Stop word ratio measured in words
	F_{58}	Diversity [20]
Punctuation	F_{59}-F_{62}	#"?", #"!" and their ratio
	F_{63}-F_{64}	#Suspension points & #Quotes
Keywords	F_{65}-F_{66}	#Comparison words (e.g. "like")
	F_{67}	#"yet" & #"sudden"
Writing Style	F_{68}-F_{69}	#All-uppercase WORDS and ratio
	F_{70}-F_{71}	#Words with repeating characters and their ratio
Text Similarity (between query& tweet)	F_{72}	Query-term proximity [2]
	F_{73}-F_{75}	#Extra/missing/overlapping terms
	F_{76}	Levenshtein distance
	F_{77}	Jaro Winkler distance
	F_{78}	Longest common subsequence
	F_{79}	Dot product
	F_{80}	Cosine similarity
	F_{81}	Jaccard sim. of unigram shingles
	F_{82}	Jaccard sim. of bigram shingles
	F_{83}	Unit match feature [21]
	F_{84}	Agreement AG (text, query) [22]
Topic	F_{85}-F_{94}	10 topics according to LDA
Word Embedding	F_{95}-F_{294}	Pre-trained Glove vectors [23]
Twitter -specific	F_{295}	#Texting lingos, e.g. haha, OMG
	F_{296}-F_{299}	#Pos./Neg. emoticons and their ratio
	F_{300}	Being retweet or not
Length	F_{301}	Tweet length ratio (in characters)
	F_{302}-F_{304}	#Words

E. Label distributions

For the classification experiments, it is necessary to consider the distribution of the sentiment labels which are shown in Fig. 2 and Fig. 3 respectively. In the former, four votes per tweet are used for the three crowdsourced corpora while all votes per tweet in S_0 are utilized. S_0 exhibits a similarly skewed label distribution as the three crowdsourced corpora, thus S_0 is representative. In all corpora similar patterns emerge in that the majority of tweets is considered *negative* while

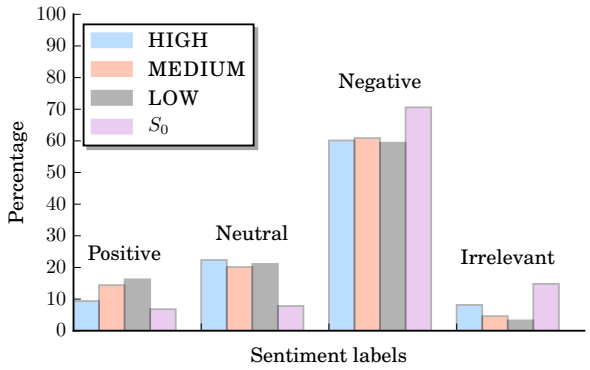

Fig. 2. Label distribution across all four labeled corpora - three crowdsourced corpora using four votes per tweet and the seed set using all votes.

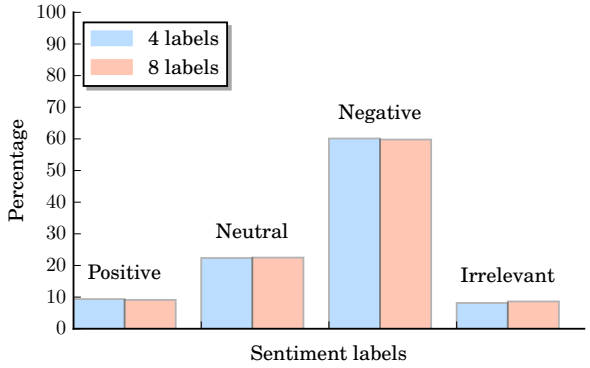

Fig. 3. Label distribution in HIGH when computing majority labels using four or eight votes per tweet.

only a few tweets are *irrelevant*. Since the three crowdsourced corpora appear internally consistent, we interpret this as a hint toward the reliability of the labels. To see how the label distribution is affected if more budget is allocated to tweets, we show the resulting distribution in Fig. 3 for HIGH according to majority voting using four and eight votes respectively. Despite increasing the number of votes, the distribution remains almost identical. We interpret this as another clue that crowd workers were honest.

V. Experimental evaluation

We examine the research questions described in Table II. While the first two research questions deal with the devised predictor, the last two questions examine the overall potential of our approach given that it is feasible to predict worker disagreement.

A. Analyzing the appropriateness of Definition 1

Before performing the actual experiments, we investigate how well Definition 1 captures the notion of *ambiguous* tweets to ensure that the findings of our experiments are valid. Therefore, we create a ground truth for TRAIN, LOW, MEDIUM, and HIGH and compare these labels with those derived from Definition 1. After a manual inspection of all

TABLE II
Overview of the research questions to be analyzed.

No.	Research Question Description
RQ_0^{1a}	DAP_0 trained on S_0 can separate *ambiguous* tweets from *unambiguous* ones.
RQ_0^{1b}	The worker disagreement in HIGH is higher than in MEDIUM and LOW.
RQ_0^2	DAP_{i+1} shows better performance than DAP_i
RQ_0^3	Removing tweets with disagreement from the training set improves predictor performance.
RQ_0^4	Allocating more budget (to recruit more workers) to tweets with disagreement does not resolve worker disagreement.

3.5k tweets, we identified four main sources that could induce high worker disagreement. When including one additional marker for tweets which do not exhibit any of these characteristics, we end up with the following five classes:

- (A)mbiguity: a tweet is difficult because it either contains mixed sentiment for one or multiple entities or the sentiment could be interpreted in different ways. Example: "I keep thinking Trump's winning, but he's also kinda acting like a clown so idk... #debatenight"
- Lack of (B)ackground knowledge: a tweet is difficult because it requires background knowledge, either in the sense of semantics, e.g. unknown entities like people or events in a tweet, or due to the lack of context. Example: "If I could ask the presidential candidates one question tonight, it would be "Would there be justice for Harambe?" #debates"
- (I)rrelevance: a tweet is difficult to label because it is irrelevant to the subject matter, e.g. a tweet that praises the clothing of the moderator. Example: ""I wait for the Lord, my whole being waits, and in His word I put my hope." Psalm 139:5 #debatenight"
- (O)ther: a tweet that is difficult to label for other reasons, i.e. it is relevant to the subject matter but it is not possible to infer what the author wants to say, e.g. due to sarcasm. Example: "I can't take either seriously until Lester Holt asks the real question in this debate: is a hot dog a sandwich? #debatenight #teachthetruth"
- (S)implicity: tweets which do not include any of the disagreement indicators. Example: "The fact that Trump cuts Lester off every time he asks a question goes to show that he has no respect for people #debatenight"

Two of the authors labeled all tweets independently in terms of these five classes. Afterwards the labels were merged in case of agreement and otherwise the authors discussed to choose a label unanimously. The resulting label distribution is visualized in Fig. 4 and suggests that most tweets are straightforward to label, while the four disagreement sources are roughly equally distributed. Since A, B, I, O indicate *ambiguous* tweets, we aggregate them into *ambiguous*. while S indicates *unambiguous* tweets. It turns out that 327/1106 *ambiguous* tweets according to Definition 1 are considered as *unambiguous* by the ground truth. One possible explana-

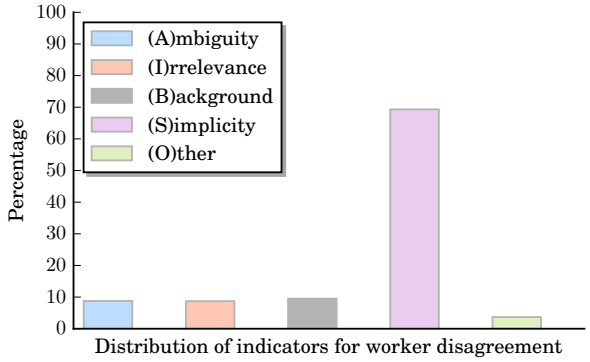

Fig. 4. Distribution of the indicators inducing worker disagreement across 3.5k tweets.

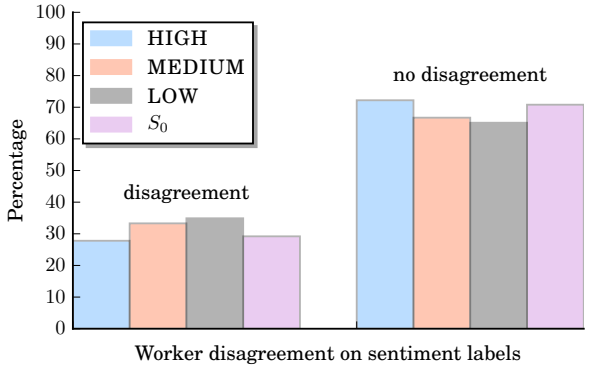

Fig. 5. Worker disagreement distributions across all four labeled corpora - three crowdsourced corpora using four votes per tweet and the seed set using all votes.

tion for the differences could be that some crowd workers assigned low-quality labels. In terms of *unambiguous* tweets according to Definition 1, the ground truth considers 295/2394 *unambiguous* tweets as *ambiguous*. This suggests that crowd workers performed more reliably on these tweets. Nevertheless, overall our analysis suggests that Definition 1 captures the difference between *ambiguous* and *unambiguous* tweets sufficiently well.

B. Q1: How Does the Disagreement Predictor Perform?

For analyzing RQ_0^{1a}, we use area under the ROC curve (AUC) which takes the skewness of the data into account, hence it is a suitable metric for us (see Fig. 5). DAP_0 separates *ambiguous* from *unambiguous* tweets. As dataset we use S_0 and optimized DAP_0 for 15 min in Auto-Weka [28] using 10-fold cross-validation and averaged the AUC over five independent runs. While performing the experiment, we noticed overfitting in multiple runs, indicated by nearly perfect AUC scores. In those cases, we ignored the run and manually repeated it using Weka [29] with the optimized parameters reported by Auto-Weka. The results are shown in the first row of Table III. The averaged AUC of 0.55 indicates that DAP_0 performs slightly better than chance which partially supports RQ_0^{1a}. However, the performance could be improved by tweaking the feature space which is beyond the scope of this paper as we are mainly interested in general trends.

To analyze RQ_0^{1b}, we computed the worker disagreement according to Definition 1 for each of the three crowdsourced corpora and illustrate the disagreement distribution in Fig. 5. Four votes per tweet were used for the three crowdsourced corpora as well as all votes per tweet in S_0. It turns out that similar trends emerge in all corpora, namely workers disagree on around 30% of the tweets, which leads to a rejection of RQ_0^{1b}. In other words, DAP_0 did not learn meaningful patterns from S_0 to distinguish different levels of disagreement. However, by expanding S_0 DAP_0's performance might improve.

C. Q2: Does the disagreement predictor improve gradually?

For our proposed method to work, the most important assumption is that DAP_i improves if S_i is expanded which is

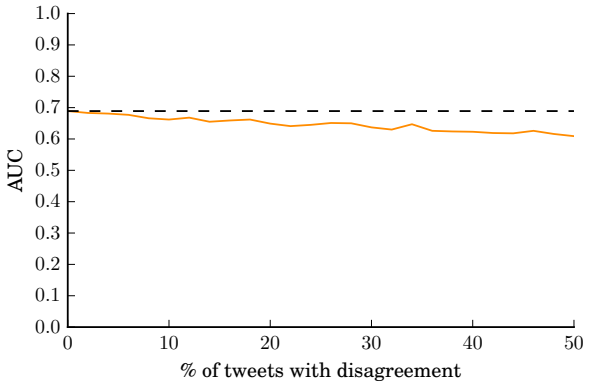

Fig. 6. Influence of tweets with disagreement on sentiment classification.

examined in RQ_0^2. We test it by comparing the performances of DAP_0 trained on S_0 and DAP_1 trained on S_1, where $S_1 = S_0 \cup \text{LOW} \cup \text{MEDIUM} \cup \text{HIGH}$. Expanding S_1 in this particular way allows us to analyze if our proposed method works in principle or not. In practice, however, S_0 should be expanded by fewer tweets at a time. Classes to be separated are the same as in Q1 – *ambiguous* and *unambiguous*. As evaluation metric we utilize AUC and we train DAP_0 and DAP_1 as described in Q1 using Auto-Weka. The results are shown in Table III. An improvement in DAP_1 over DAP_0 of 6% supports RQ_0^2 that our proposed methodology gradually refines the disagreement predictor over multiple iterations.

TABLE III
AUC SCORES OBTAINED IN FIVE AUTO-WEKA RUNS FOR DAP_0 TRAINED ON S_0 AND DAP_1 TRAINED ON S_1 RESPECTIVELY.

Run	1	2	3	4	5	Avg. AUC
DAP_0	0.57	0.57	0.47	0.57	0.57	0.55
DAP_1	0.56	0.7	0.53	0.63	0.65	0.61

D. Q3: What is the effect of disagreement on sentiment classification?

For analyzing RQ_0^3, we devise the following simulation. We use S_1 from Q2 to train STP that separates the classes *positive*, *negative*, *neutral*, and *irrelevant*. We use all votes in S_1 per tweet, i.e. all votes in S_0, LOW etc. We utilize worker disagreement according to Definition 1 to create two corpora from S_1: D containing 1.1k tweets with disagreement and ND comprising 2.2k tweets with no disagreement. That means disagreement labels are only exploited to group the tweets initially. Other than that sentiment labels are to be predicted. In the simulation, we increase the fraction of tweets with disagreement in ND by randomly choosing m tweets from ND with no disagreement and replacing them by m random tweets from D with disagreement. This way, the size of ND is fixed while the fraction of tweets with disagreement in ND increases up to 50%[7], allowing us to train multiple versions of STP on ND. We employ 10-fold cross-validation to avoid introducing any bias and we report the performance in terms of AUC averaged over three independent runs to make the results more robust. As a predictor we select a random forest and optimize it to deal with class imbalance (see Fig. 2). The reason for choosing random forest is that it is a predictor ensemble which tends to give more stable results than single predictors [30]. The result of our simulation is shown in Fig. 6 and supports RQ_0^3: STP's performance drops by up to 8% when the fraction of tweets with disagreement increases. Repeating this experiment with an unoptimized random forest predictor leads to the same result and AUC drops by up to 13%.

E. Q4: What is the effect of allocating more budget to ambiguous tweets?

To address RQ_0^4, we first analyze how worker disagreement develops when labeling budget is increased. If the labeling budget in HIGH is doubled from four to eight votes per tweet, worker disagreement decreases by 5% from 33% to 28%. This suggests that assigning more budget to ambiguous tweets can be helpful.

This is further supported by Fig. 7 in which we plotted the fraction of tweets with disagreement over all three crowdsourced corpora considering only the first n labels, where $n = 2...8$. For $n = 2...4$ we computed the disagreement for each of the three corpora, while starting from $n = 5$ only HIGH is used because the other corpora received only four votes. The plot illustrates that the valleys and peaks start to converge when increasing the number of votes. This suggests that adding more budget helps resolve some disagreement, especially if only few votes are available, but then the disagreement starts to converge and acquiring additional labels leads to diminishing returns. The valleys and troughs are most likely an artifact of our definition of majority because for an even number of votes the likelihood for worker disagreement increases as opposed to an odd number of votes.

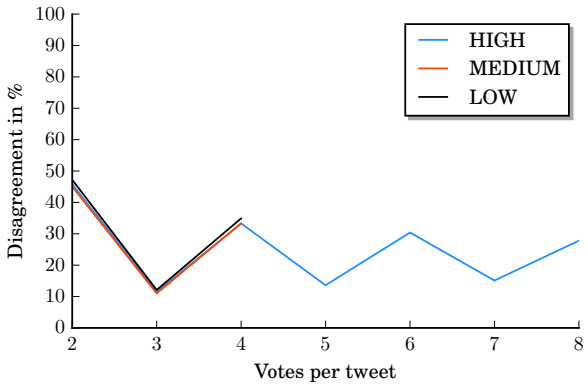

Fig. 7. Fraction of tweets with disagreement when using only the first n votes for deriving majority labels. For $n = 2, 3, 4$ we depict the fractions separately for LOW, MEDIUM, and HIGH, while for $n > 4$ only tweets from HIGH are available.

In a last step, to analyze how the performance of STP is affected by more budget allocated to tweets with disagreement, we designed another simulation similar to Q3 as follows. From HIGH we select only tweets whose agreement never changes when using the first n votes, where $n = 4...8$ to generate two corpora. This way, the same tweets are used in all runs of the experiment and only the sentiment labels of tweets with disagreement might change due to more votes. We split the tweets into ND (586 tweets) and D (87 tweets) and fix the corpus size to 174 tweets[8], initially all tweets are from ND and then we gradually replace them by tweets from D in the same manner as in Q3. The resulting performances of STP, for which we used again an optimized random forest predictor, are shown in Fig. 8. They support RQ_0^4 since the use of more votes does not improve the AUC scores. Surprisingly, contrary to RQ_0^3, STP's performance improves by 1-5% as the fraction of tweets with disagreement increases. However, repeating the experiment with an unoptimized random forest predictor supports RQ_0^4 in that more votes do not improve AUC scores and in line with Q3 the AUC drops by 4-9% when the fraction of tweets with disagreement increases. Therefore, we believe the increased AUC scores of the optimized predictor to be an artifact of the small corpus size and the randomized cross-validation splits because the other seven experiments in Q3 and Q4 using optimized and unoptimized predictors point to the opposite pattern in agreement with RQ_0^3. Overall, our results support RQ_0^4; only if tweets received less than four votes, allocating more budget to them resolves some disagreement. However, not all disagreement can be resolved which hints at aleatoric uncertainty.

[7]We obtained similar results in that the performance of STP dropped by 8% when using 1.1k tweets in ND to analyze what happens if the corpus is comprised of up to 100% tweets with disagreement. Since this scenario is less realistic, we do not depict the results.

[8]Repeating the experiment with the same settings as in Q3, now using only 87 tweets instead of 174 tweets in ND (which leads to up to 100% of tweets with disagreement), we observe a drop in STP's AUC by 2-6% and more votes per tweet do not remedy these drops.

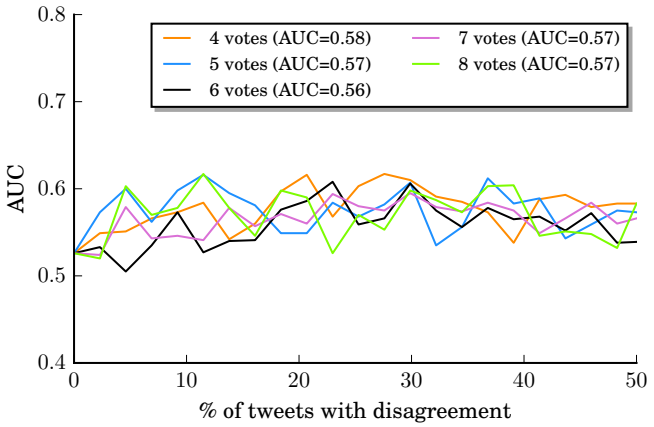

Fig. 8. Influence of tweets with disagreement on the predictor performance if the number of votes used for majority voting increases. The AUC scores in the legend are averaged per curve.

VI. Discussion and conclusion

In this study, we first investigated whether disagreement among the labels assigned to tweets by crowd workers can indeed be alleviated by acquiring more labels. We designed an iterative process that involves disagreement prediction and uses polarity classification as the crowd labeling task. We have shown experimentally that disagreement among the labels assigned to tweets by crowd workers impacts polarity classification quality negatively. This finding agrees with earlier studies on the behavior of crowd workers. However, our results also indicate that such a disagreement cannot be always alleviated by acquiring more labels for the tweets, for which disagreement occurs. Indeed, Fig. 7 shows that as votes (labels for tweets) are added, the disagreement oscillates instead of converging fast towards zero. The slow shift to lower levels of oscillation implies that for some tweets it is beneficial to add more labels, but not for all of them because some tweets are inherently controversial. We expect that acquiring more labels for tweets with disagreement is only beneficial if tweets have few votes. Otherwise the additional labeling costs outweigh the reduced worker disagreement. However, finding the optimal trade-off between removing tweets and allocating more budget to them is future work.

Our iterative process allows the experiment designer to allocate crowd workers for fractions of the unlabeled corpus, so that the amount of disagreement is monitored. Our results show that our disagreement predictor separates between tweets with and without disagreement to some extent, and that it improves as it sees more labeled data. Hence, the experimenter can stop the crowd labeling process when the predictor converged and then decide how the disagreement tweets should be treated, while the no disagreement tweets are given to the crowd workers. Nevertheless, we plan to experiment with different tweet representations like [31] to improve the performance of the disagreement predictor. Another potential avenue for identifying a better feature space for the disagreement predictor is indirectly described in Section V-A as we identified four main sources that induce crowd worker disagreement. Extracting more features related to these sources seems promising. Furthermore, analyzing *why* crowd workers consider certain tweets as *ambiguous* in contrast to the ground truth and vice versa is worth more research. This way one could tease apart aleatoric and epistemic uncertainty. Another possible outcome from such an analysis could be a more suitable definition of worker disagreement as Definition 1 becomes less reliable for *ambiguous* tweets with a discrepancy of 29.5% between crowd workers and the ground truth. Multiple factors could account for this to some extent, e.g. low-quality labels or aleatoric uncertainty. However, perhaps this observation indicates that *ambiguous* tweets should not be labeled by crowd workers, but experts instead if one requires reliable labels. Especially analyzing why some tweets are considered *unambiguous* by crowd workers but not experts demands a detailed analysis, e.g. workers might agree due to chance as they employ similar backup strategies in case of uncertainty like assigning *neutral* sentiment. Being able to identify and prevent such situations would improve label quality. One idea for an alternative definition of worker disagreement would be quantifying a majority label in terms of the difference, epsilon, between the most frequent and second most frequent label. Then a tweet is considered *ambiguous* if the actual difference between those labels is smaller than epsilon, where epsilon could be a constant or a relative number, e.g. twice as much as the least frequently chosen label.

Our finding on the unresolvable disagreement for some tweets has implications on the design of crowdsourcing experiments. Although such experiments are often very well-designed, it is possible that the set of labels needed to characterize the tweets must be larger or different than the one originally anticipated, e.g. to accommodate a label ”controversial” or ”bipolar”. Our iterative methodology allows the experimenter to identify such a phenomenon at an early iteration, before using up the whole budget.

While our proposed crowdsourcing methodology is applicable to different fields such as text or image analysis, our features proposed in Section IV-D are text-related, meaning that one would have to derive different features when dealing with inputs other than text. A further shortcoming of our findings concerns the convergence of the disagreement predictor: in each iteration, it assigns labels without learning from past misclassifications. We intend to replace this predictor by an incremental one, to ensure faster convergence. We also plan to investigate the relationship between convergence speed and budget usage, which here translates to the number of tweets being labeled at each iteration.

A further limitation of our findings concerns the separation between disagreement due to internal features of the tweets and disagreement due to features of the crowd workers. The oscillation of disagreement indicates the presence of such internal features, while the reduction of disagreement indicates the influence of the crowd workers themselves. A step towards discerning the two aspects is the inspection of the tweets, but

this is a strenuous, non-automated step. However, our approach of measuring disagreement over time can help an experimenter *see* the impact of more labels on the agreement oscillation, as it was shown here in Figure 7. By fitting a line to the oscillating curve and computing the slope of this line, we may provide an estimate of convergence. In this work, we have studied the oscillation in one experiment; more experiments on different corpora are needed to understand when and how the disagreement may converge.

Our tweet corpus has been built on the basis of keywords. It is likely that some tweet collections contain less disagreement-provoking tweets. Hence, we plan to run our experiments on more collections, with different keywords, and seek to identify features that are predictive of disagreement. Nonetheless, disagreement does show up in crowd labeling experiments. We have shown that our methodology helps in identifying it. Our dataset[9] and source code[10] are both publicly available.

REFERENCES

[1] S. Räbiger, M. Spliliopoulou, and Y. Saygın, "How do annotators label short texts? toward understanding the temporal dynamics of tweet labeling," *Information Sciences*, vol. 457-458, pp. 29 – 47, 2018.

[2] A. C. Williams, J. Goh, C. G. Willis, A. M. Ellison, J. H. Brusuelas, C. C. Davis, and E. Law, "Deja vu: Characterizing worker reliability using task consistency," in *Proceedings of the 5th AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*, 2017, pp. 197–205.

[3] T. McDonnell, M. Lease, T. Elsayad, and M. Kutlu, "Why is that relevant? collecting annotator rationales for relevance judgments," in *Proceedings of the 4th AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*, 2016, pp. 139–148.

[4] D. Dunning, "The dunning–kruger effect: On being ignorant of one's own ignorance," in *Advances in experimental social psychology*. Elsevier, 2011, vol. 44, pp. 247–296.

[5] U. Gadiraju, J. Yang, and A. Bozzon, "Clarity is a worthwhile quality: On the role of task clarity in microtask crowdsourcing," in *Proceedings of the 28th ACM Conference on Hypertext and Social Media*. ACM, 2017, pp. 5–14.

[6] J. Whitehill, T. fan Wu, J. Bergsma, J. R. Movellan, and P. L. Ruvolo, "Whose vote should count more: Optimal integration of labels from labelers of unknown expertise," in *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, Eds. Curran Associates, Inc., 2009, pp. 2035–2043.

[7] R. Senge, S. Bösner, K. Dembczyński, J. Haasenritter, O. Hirsch, N. Donner-Banzhoff, and E. Hüllermeier, "Reliable classification: Learning classifiers that distinguish aleatoric and epistemic uncertainty," *Information Sciences*, vol. 255, pp. 16–29, 2014.

[8] D. Chandler and A. Kapelner, "Breaking monotony with meaning: motivation in crowdsourcing markets," *Journal of Economic Behavior & Organization*, vol. 90, pp. 123–133, 2013.

[9] G. Kazai, J. Kamps, and N. Milic-Frayling, "An analysis of human factors and label accuracy in crowdsourcing relevance judgments," *Information retrieval*, vol. 16, no. 2, pp. 138–178, 2013.

[10] U. Gadiraju, B. Fetahu, and R. Kawase, "Training workers for improving performance in crowdsourcing microtasks," in *Design for Teaching and Learning in a Networked World*. Springer, 2015, pp. 100–114.

[11] D. Zhu and B. Carterette, "An analysis of assessor behavior in crowdsourced preference judgments," in *SIGIR 2010 workshop on crowdsourcing for search evaluation*, 2010, pp. 17–20.

[12] L. Aroyo and C. Welty, "Crowd truth: Harnessing disagreement in crowdsourcing a relation extraction gold standard," *WebSci2013. ACM*, vol. 2013, 2013.

[13] H. M. Alonso, A. Johannsen, O. L. de Lacalle, and E. Agirre, "Predicting word sense annotation agreement," in *Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics (LSDSem)*, 2015, p. 89.

[14] P. S. Bayerl and K. I. Paul, "Identifying sources of disagreement: Generalizability theory in manual annotation studies," *Computational Linguistics*, vol. 33, no. 1, pp. 3–8, 2007.

[15] T. Mitra, C. J. Hutto, and E. Gilbert, "Comparing person-and process-centric strategies for obtaining quality data on amazon mechanical turk," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2015, pp. 1345–1354.

[16] S. Doroudi, E. Kamar, E. Brunskill, and E. Horvitz, "Toward a learning science for complex crowdsourcing tasks," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 2016, pp. 2623–2634.

[17] D. Reidsma and J. Carletta, "Reliability measurement without limits," *Computational Linguistics*, vol. 34, no. 3, pp. 319–326, 2008.

[18] X. Gan, X. Wang, W. Niu, G. Hang, X. Tian, X. Wang, and J. Xu, "Incentivize multi-class crowd labeling under budget constraint," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 4, pp. 893–905, 2017.

[19] M. Sameki, M. Gentil, K. K. Mays, L. Guo, and M. Betke, "Dynamic allocation of crowd contributions for sentiment analysis during the 2016 us presidential election," *arXiv preprint arXiv:1608.08953*, 2016.

[20] K. Tao, F. Abel, C. Hauff, and G.-J. Houben, "What makes a tweet relevant for a topic?" in *# MSM*, 2012, pp. 49–56.

[21] A. Dong, R. Zhang, P. Kolari, J. Bai, F. Diaz, Y. Chang, Z. Zheng, and H. Zha, "Time is of the essence: improving recency ranking using twitter data," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 331–340.

[22] S. Ravikumar, K. Talamadupula, R. Balakrishnan, and S. Kambhampati, "Raprop: Ranking tweets by exploiting the tweet/user/web ecosystem and inter-tweet agreement," in *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. ACM, 2013, pp. 2345–2350.

[23] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[24] S. Baccianella, A. Esuli, and F. Sebastiani, "Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining," in *in Proc. of LREC*, 2010.

[25] S. Rosenthal, P. Nakov, S. Kiritchenko, S. Mohammad, A. Ritter, and V. Stoyanov, "Semeval-2015 task 10: Sentiment analysis in twitter," in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Denver, Colorado: Association for Computational Linguistics, June 2015, pp. 451–463.

[26] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2004, pp. 168–177.

[27] D. M. Blei, A. Y. Ng, M. I. Jordan, and J. Lafferty, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, p. 2003, 2003.

[28] L. Kotthoff, C. Thornton, H. H. Hoos, F. Hutter, and K. Leyton-Brown, "Auto-weka 2.0: Automatic model selection and hyperparameter optimization in weka," *Journal of Machine Learning Research*, vol. 17, pp. 1–5, 2016.

[29] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[30] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*. Springer, 2000, pp. 1–15.

[31] S. Vosoughi, P. Vijayaraghavan, and D. Roy, "Tweet2vec: Learning tweet embeddings using character-level cnn-lstm encoder-decoder," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 2016, pp. 1041–1044.

[9] https://www.researchgate.net/publication/326625792_Dataset_for_our_paper_titled_Predicting_worker_disagreement_for_more_effective_crowd_labeling

[10] https://github.com/fensta/DSAA2018