ANALYZING CROWD WORKERS' LEARNING BEHAVIOR TO OBTAIN MORE RELIABLE LABELS

by Stefan Räbiger

Submitted to the Graduate School of Engineering and
Natural Sciences
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Sabancı University
August, 2018

ANALYZING CROWD WORKERS' LEARNING BEHAVIOR TO OBTAIN MORE RELIABLE LABELS

APPROVED BY:

Professor Yücel Saygın ..... (Thesis Advisor)

Assistant Professor Kamer Kaya .....

Associate Professor Kemal Kılıç .....

Associate Professor Emine Yılmaz .....

Associate Professor A. Şima Etaner-Uyar .....

DATE OF APPROVAL .....

© Stefan Räbiger 2018
All Rights Reserved

Flower gardens are awesome

Acknowledgments

This thesis would not have been possible without the guidance and expertise of my thesis advisor, Prof. Yücel Saygın and my co-advisor, Prof. Myra Spiliopoulou. Their constructive feedback and ideas helped shape this thesis. I am also grateful for the helpful comments of the other jury members, Asst. Prof. Kamer Kaya, Assoc. Prof. Kemal Kılıç, Assoc. Prof. Emine Yılmaz, and Assoc. Prof. A. Şima Etaner-Uyar. Thanks a lot to my muse Ece Tarhan who has piqued my curiosity about so many (not necessarily thesis-related) topics, and for inspiring me. I am happy to have crossed paths with Gülbostan and Anar Abliz as they are genuinely wonderful people and true friends. I also appreciate the support of my colleague and friend Gizem Gezici for her translation of the Turkish abstract and for helping me solve some thesis-related problems despite her busy schedule. My thanks also go to my friends Mike Stephan, Rene Müller, and Christian Beyer. Similarly, Hüseyin Tufan Usta and the rest of the Sabancı badminton team deserve a shout-out for giving me the opportunity to find a place for some work-life balance while honing my skills thanks to Beyhan Özgür significantly (in both the statistical and non-statistical sense). Of course, I also want to express my gratitude to my mom, Christiane Räbiger, and my dad, Hartwig Räbiger, for their unconditional support over the years - be it financially which allowed me to focus on my studies, be it morally, or through sending me small items from Germany etc. Thanks to my brother, Michael Räbiger, for distracting me successfully from my work and to FromSoftware for creating a wonderfully addictive fantasy trilogy that served well as an example in this thesis and for training my patience.

I would also like to thank all volunteers in Sabancı and Magdeburg for their participation in our labeling experiment which formed the basis for this thesis.

ANALYZING CROWD WORKERS' LEARNING BEHAVIOR TO OBTAIN MORE RELIABLE LABELS

Stefan Räbiger

Computer Science and Engineering

Ph.D. Thesis, 2018

Thesis Supervisor: Prof. Yücel Saygın

Thesis Co-supervisor: Prof. Myra Spiliopoulou

Keywords: worker disagreement, crowdsourcing, dataset quality, label reliability, tweet ambiguity, annotation behavior, annotation behavior, learning effect, human factors

Abstract

Crowdsourcing is a popular means to obtain high-quality labels for datasets at moderate costs. These crowdsourced datasets are then used for training supervised or semi-supervised predictors. This implies that the performance of the resulting predictors depends on the quality/reliability of the labels that crowd workers assigned – low reliability usually leads to poorly performing predictors. In practice, label reliability in crowdsourced datasets varies substantially depending on multiple factors such as the difficulty of the labeling task at hand, the characteristics and motivation of the participating crowd workers, or the difficulty of the documents to be labeled. Different approaches exist to mitigate the effects of the aforementioned factors, for example by identifying spammers based on their annotation times and removing their submitted labels.

To complement existing approaches for improving label reliability in crowdsourcing, this thesis explores label reliability from two perspectives: first, how the label reliability of crowd workers develops over time during an actual labeling task, and second how it is affected by the difficulty of the documents to be labeled.

We find that label reliability of crowd workers increases after they labeled a certain number of documents. Motivated by our finding that the label reliability for more difficult documents is lower, we propose a new crowdsourcing methodology to improve label reliability: given an unlabeled dataset to be crowdsourced, we first train a difficulty predictor

on a small seed set and the predictor then estimates the difficulty level in the remaining unlabeled documents. This procedure might be repeated multiple times until the performance of the difficulty predictor is sufficient. Ultimately, difficult documents are separated from the rest, so that only the latter documents are crowdsourced. Our experiments demonstrate the feasibility of this method.

Ķitle Çalışanlarının Öğrenme Tutumlarının Daha Güvenilir Etiketler Elde Etmek İçin Analiz Edilmesi

Stefan Räbiger

Bilgisayar Bilimi ve Mühendisliği

Doktora Tezi, 2018

Tez Danışmanı: Prof. Dr. Yücel Saygın

Eş-Tez Danışmanı: Prof. Dr. Myra Spiliopoulou

Anahtar Sözcükler: çalışan anlaşmazlığı, kitle-kaynak yöntemi, veri seti kalitesi, etiket güvenirliliği, tweet anlam belirsizliği, etiketleme tutumu, öğrenme etkisi, insan faktörleri

Özet

Kitle-kaynak, veri kümeleri için yüksek kaliteli etiketleri makul maliyetler ile elde etmek için kullanılan popüler bir yöntemdir. Bu kitle-kaynak yöntemiyle etiketlenen veri setleri, sonrasında gözetimli veya yarı-gözetimli sınıflayıcıların eğitimi için kullanılır. Bu da, bu prosedür sonucunda oluşan sınıflayıcı performanslarının kitle çalışanlarının atadığı etiketlerin kalitesi/güvenirliliğine bağlı olduğu anlamına gelmektedir - düşük güvenirlilik genellikle yetersiz çalışan sınıflayıcılara sebep olur. Pratikte, kitle-kaynak veri kümelerindeki etiket güvenirliliği, eldeki etiketleme işinin zorluğu, katılımcı kitle çalışanlarının özellikleri ve motivasyonu, veya etiketlenecek dokümanların zorluğu gibi birçok faktöre bağlı olarak büyük ölçüde değişkenlik gösterir. Bu bahsedilen faktörlerin etiketlerin kalitesine etkisini hafifletmek için ise, verilen kitle-kaynak görevini tanımına uygun olarak yerine getirmeyen (spammer) çalışanları, etiketleme sürelerine bakarak belirlemek ve gönderdikleri etiketleri silmek gibi farklı yaklaşımlar mevcuttur.

Bu tez, kitle-kaynak yönteminden elde edilen etiket güvenirliliğini iyileştirerek mevcut yaklaşımları tamamlamak amacıyla, etiket güvenirliliği konusunu ilk olarak, gerçek bir etiketleme işi süresince kitle çalışanlarının etiket güvenirliliğinin zamanla nasıl geliştiği, ve ikinci olarak etiketlerin etiketlenecek dokümanların zorluğundan nasıl etkilendiği olmak üzere iki açıdan incelemektedir.

Kitle-kaynak yöntemi ile etiketlenen veri seti üzerinde yaptığımız analizler sonucunda, kitle çalışanlarının etiket güvenirliliğinin belli sayıda dokümanı etiketledikten sonra arttığını gözlemledik. Bunun sonucunda ve daha zor dokümanlar için etiket güvenirliliğinin daha düşük olması bulgusundan yola çıkarak, etiket güvenirliliğini iyileştirmek için yeni bir kitle-kaynak yöntembilimi önermekteyiz. Önerdiğimiz bu metodolojide, kitle-kaynak yöntemiyle etiketlenecek olan elimizdeki etiketsiz veri setini kullanarak, öncelikle küçük bir başlangıç seti üzerinde bir zorluk tahmin edici (predictor) eğitip, sonrasında bu tahmin ediciden yararlanarak başlangıç seti dışında kalan dokümanların zorluk derecesini tahmin etmeyi hedefliyoruz. Bu prosedür, eğitilen tahmin edicinin performansı yeterli seviyeye ulaşana kadar birçok kez tekrarlanabilir. Son olarak, bu adımlar sonucunda elde edilen tahmin edici kullanılarak tespit edilen zor dokümanlar, veri setinin geri kalanından ayrılır ve sadece bu veri kümesinde kalan dokümanlar kitle-kaynak yöntemi ile etiketlenir. Deney sonuçlarımız da, bu yöntemin kitle-kaynak yöntemi ile elde edilen etiketlerin güvenirliliği üzerinde etkili olduğunu göstermektedir.

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations and Symbols

TRAIN	Dataset containing 500 tweets
TRAIN$_S$	Dataset containing 200 randomly selected tweets from TRAIN
C	Dataset containing 19.5k tweets
U	TRAIN $\cup$ C, i.e. 20k tweets
R	All tweets that will not be included in C as they are predicted by DAP to have *Disagreement*
DAP	Disagreement predictor
STP	Sentiment predictor
SU	Sabancı University (Turkey)
MD	University of Magdeburg (Germany)
S	Crowd workers in this group labeled 50 tweets from TRAIN
M	Crowd workers in this group labeled 150 tweets from TRAIN
L	Crowd workers in this group labeled 500 tweets from TRAIN
AL	Active (machine) learning
$NormSim$	Normalized similarity between two tweets

Chapter 1

Introduction

Crowdsourcing is a popular means to obtain high-quality labels with a limited budget. In crowdsourcing non-experts, so called crowd workers, complete micro-tasks, in which they label small subsets of the whole dataset. For each completed micro-task they receive a monetary compensation. The central idea of crowdsourcing is that multiple cheap crowd workers assign a label to each document instead of requesting expensive experts to assign a single label to all documents. As a result, datasets are faster labeled with crowdsourcing as more workers than experts are available. Moreover, the monetary compensation for crowd workers is substantially lower than for experts. Typically, a single expert assigns a label to a document, which makes it automatically the final label (ground truth). However, multiple labels exist per document (assigned by multiple crowd workers) in crowdsourcing as crowd workers lack background knowledge. Thus, the labels must be aggregated to single labels because, ultimately, this ground truth will be used for training supervised and semi-supervised predictors. Multiple experiments, e.g. [1, 2], have demonstrated the potential of crowdsourcing in that the ”wisdom of the crowd” effect, i.e. the aggregated labels of multiple workers, rivals the quality of expert labels despite crowd workers usually lacking background knowledge.

A typical crowdsourcing workflow is depicted in Figure 1.1. In the first step, the requester[1] designs the labeling task for the crowdsourcing platform, e.g. Amazon Me-

[1]In this thesis, we refer to a requester as an *experimenter* to highlight her role. An experimenter is a

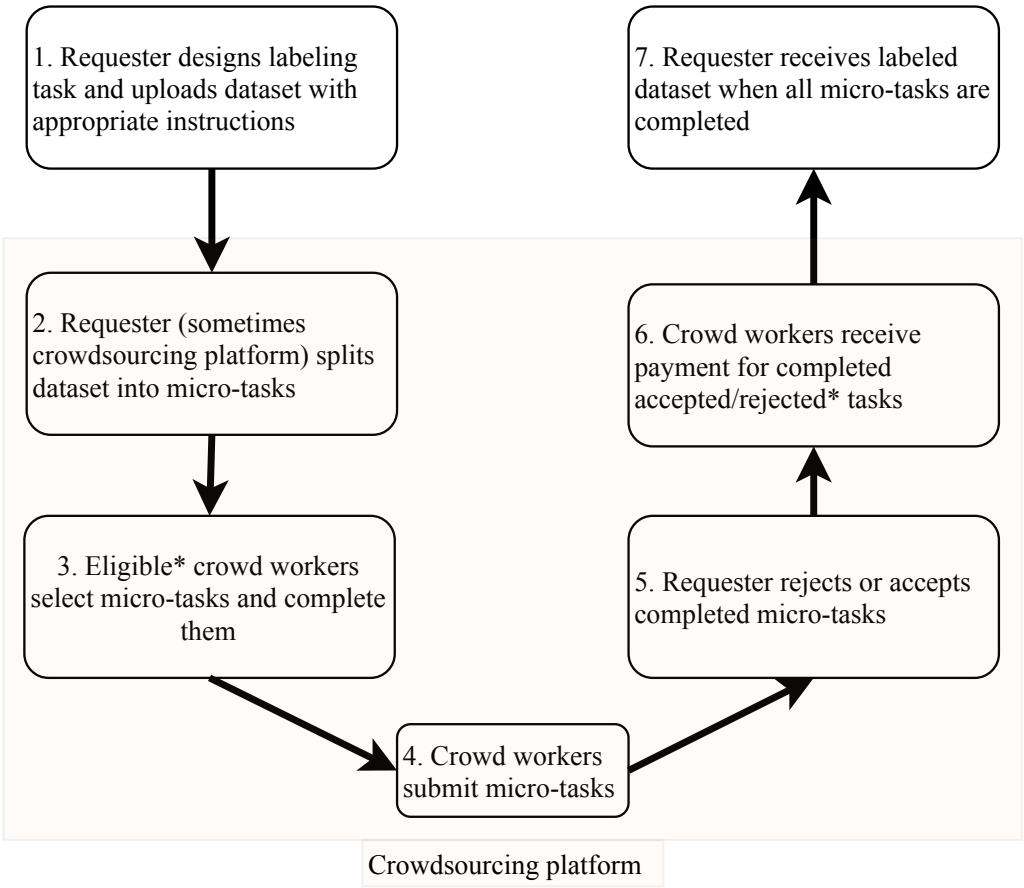


Figure 1.1: Schematic illustration of a typical crowdsourcing workflow.

chanical Turk, to which she will upload the dataset to be labeled. This design process involves creating instructions for crowd workers and deciding about how many documents are contained in a micro-task, which contains a subset of the documents to be labeled. Furthermore, the experimenter sets the payment per completed micro-task and whether crowd workers should always be paid even if their submitted micro-tasks were deemed inaccurate by the experimenter. Last but not least, the experimenter also identifies quality criteria that interested crowd workers must meet before they are allowed to complete any of these available micro-tasks. The requester might decide to split the unlabeled dataset into micro-tasks herself and upload these instead of the dataset to the crowdsourcing plat-

special kind of requester, namely a person conducting crowdsourcing experiments for research.

form in the second step. Alternatively, the crowdsourcing platform, assuming it provides this feature, might perform this task of creating the micro-tasks from the uploaded dataset. In the third step, only workers who meet the predefined quality criteria, that were defined by the experimenter, are able to complete any of the available micro-tasks. In steps four and five workers submit their micro-tasks upon completion and receive their payments. Afterwards, in step seven, once all documents are labeled, the experimenter receives the fully labeled dataset.

1.1 Motivation

Despite the popularity of crowdsourcing as a means to obtain large-scale, labeled datasets, the quality of the datasets largely varies because the label reliability, that is the reliability of the labels that crowd workers assigned, is unknown. This is problematic because training predictors relies on a reliable ground truth – otherwise the resulting predictors might poorly estimate the labels of unlabeled documents. This could happen as the predictor might be unable to extract relevant patterns from training documents which were assigned unreliable labels by crowd workers. For example, suppose one wants to train a predictor that distinguishes the sentiment (positive, neutral, negative) of tweets. Crowd workers assigned each tweet one of the labels ”positive”, ”neutral”, or ”negative”. However, if they chose the wrong one for some reasons, the predictor might miss important patterns because during the training procedure it does not have access to all truly ”positive”, ”neutral”, and ”negative” tweets to learn the differences between all three labels.

There are many reasons why crowd workers assign wrong labels to documents. For one, crowdsourcing is most reliable in situations where a correct answer to a question exists [3]. Moreover, low-quality workers like spammers, inexperienced and/or unmotivated workers [4, 5, 6] are attracted to crowdsourcing platforms as workers are compensated with monetary rewards for completing labeling tasks. One countermeasure to remove any labels submitted by such low-quality workers is to leverage human factors, i.e. traits of crowd workers, to analyze which worker characteristics are important for

acquiring reliable labels. Examples for human factors include patterns in annotation behavior [7] to identify spammers, the level of expertise [8], age of workers [9], and many others. Besides human factors, which only consider worker-related factors, there are also task-related factors, e.g. a clear task specification contributes to more reliable labels [10], and document-related factors like document difficulty [11] that affect label reliability.

It is difficult for experimenters to take all of these factors into consideration when designing the labeling task because the analysis of these factors depends on the metadata provided by crowdsourcing platforms – if a platform does not provide specific metadata, the respective factor cannot be considered for determining the label reliability. To avoid this limitation, one could implement one's own crowdsourcing platform, but this implementation will take time to get adopted by requesters and crowd workers in the best case. In the worst case, this new implementation will be ignored. Therefore, it is more promising for experimenters to use one of the popular crowdsourcing platforms like Amazon Mechanical Turk or CrowdFlower which have a large number of crowd workers. This dependency of experimenters on the metadata provided by existing crowdsourcing platforms limits the quality of crowdsourced datasets. Hence, it is desirable to identify methods enhancing label reliability that are independent of the underlying crowdsourcing platform. Translated to Figure 1.1, this means that methods to enhance label reliability should be applied either in step one or seven. The goal of this thesis is to propose such a method for step one, i.e. before the dataset is sent to a crowdsourcing platform. The only type of metadata that would be leveraged by our proposed method during preliminary experiments is the annotation time of each document, i.e. how long it took a worker to assign a label to the document, to determine the length of a worker's learning process. Fortunately, this metadata is provided by popular crowdsourcing platforms like Amazon Mechanical Turk and CrowdFlower by default.

1.2 Thesis Scope and Research Questions

One neglected aspect in the discussion of label reliability in crowdsourcing is that crowd workers gradually acquire experience and background knowledge in a labeling task over time. In other words, they undergo a learning process. Thus, one would expect them to become more accurate at assigning labels to documents over time. For example, if the task is about assigning sentiment labels to tweets involving "Dark Souls III", workers might initially know little about the topic. However, after reading and labeling some tweets, they realize that it is a challenging role-playing computer game in a Dystopian world, they also notice common key words that identify the dominant sentiment in a tweet, etc. To the best of our knowledge, no one has analyzed the connection between label reliability and the learning process. Another aspect that deserves consideration in this analysis is the effect of document difficulty on label reliability. Intuitively, one would expect labels for difficult documents to be less reliable. But is this assumption true? Or is it affected by the learning process? Therefore, this thesis also studies how document difficulty affects label reliability. Last but not least, if the difficulty of documents potentially affects label reliability, it seems promising to identify these difficult documents and separate them from the rest to improve label reliability in crowdsourcing. That is the key idea of our proposed crowdsourcing methodology.

In this thesis, we use a hierarchical sentiment labeling task on Twitter as the acquisition of reliably labeled texts is a challenge, because tweets are posted continuously and exhibit great diversity in language and content. Moreover, sentiment analysis is known to be subjective and therefore sufficiently difficult. This difficulty is also perceived by crowd workers [12], enforcing them to learn over time how to assess sentiment more accurately. As topic for the sentiment analysis, we focus on tweets that were published during the first debate between Hillary Clinton and Donald Trump during the US presidential election 2016. Choosing such a hot, polarizing topic increases the chances of encountering difficult tweets which we require for our analysis. In light of the above problems, this thesis answers the following research questions:

- **RQ1**. How does labeling behavior of crowd workers over time affect their label reliability?
- **RQ2**. How does tweet difficulty affect the label reliability of crowd workers over time?
- **RQ3**. Can we improve label reliability by utilizing findings from RQ1 and RQ2?

1.3 Contributions

It is known that crowd workers undergo a learning process, i.e. their annotation times initially drop rapidly and then they converge to a stable level [13, 7]. We refer to the early phase as learning phase and to the late phase as exploitation phase. The main contributions of this thesis are:

1. We find that the label reliability of crowd workers is lower in the learning phase than in the exploitation phase (Chapter 4).
2. We quantify the length of a crowd worker's learning phase in terms of how many documents she labeled before which helps estimating a worker's label reliability (Chapter 4).
3. We discover that document difficulty affects the label reliability of a crowd worker in the exploitation phase negatively, while no effect can be observed in the learning phase (Chapter 5).
4. We propose a workflow that filters out such difficult documents before crowdsourcing the remaining documents (Chapter 6).
5. We create labeled benchmark datasets for sentiment analysis[2] (Chapter 4) and document difficulty [3] (Chapter 6) to help other researchers investigate document difficulty.

[2] https://www.researchgate.net/publication/325180810_Infsci2017_dataset

[3] https://www.researchgate.net/publication/326625792_Dataset_for_our_

1.4 Thesis Outline

The overall goal of this thesis is to increase the reliability of crowdsourced datasets as motivated in this chapter. After discussing existing literature in the field in Chapter 2, we describe the Twitter dataset to be used throughout this thesis in Chapter 3 as well as introduce fundamental concepts. Chapter 4 addresses RQ1 by analyzing the behavior of crowd workers while they complete the sentiment labeling task. Chapter 5 builds on these findings to examine RQ2, that is how tweet difficulty influences this label reliability of crowd workers. The findings from Chapter 4 and 5 motivate a new crowdsourcing methodology that is described in Chapter6, where we try to predict the difficulty of tweets to answer RQ3. Chapter 7 concludes this thesis by summing up the main ideas and discussing potential implications and applications of our findings including avenues for future research.

---

paper_titled_Predicting_worker_disagreement_for_more_effective_crowd_labeling

Chapter 2

Related Work

Although crowdsourcing has many benefits, it provides an uncontrolled environment [14]: ” As the entire [crowdsourcing] process, such as recruitment, task assignment and result collection, is done on the Internet, the requester will not get a chance to meet any worker. Hence, the requester will not know whether a worker is genuine or a spammer as he or she does not have access to their personality data. ” This implies that low-quality workers exist who assign unreliable labels. Thus it is crucial to identify them and remove their submitted micro-tasks. We therefore discuss multiple indicators of crowd workers that suggest good/bad worker performance and focus specifically on annotation time as this is the aspect we use in this thesis. Similarly, we review literature that models document difficulty, and in particular tweet difficulty in crowdsourcing and similar environments. We also discuss in the context of crowdsourcing how worker disagreement on a document is utilized to estimate the document's difficulty.

While most of the work we discuss is about the domain crowdsourcing, some studies come from the domain active machine learning[1] [15]. Those fields differ in their objectives, but the quality of the labels obtained from the workers is mission-critical in both fields.

[1]We use this term instead of the more common one *active learning* to emphasize that we mean active learning in machine learning and not students participating more actively in the learning process.

2.1 Human Factors

Several human factors, which denote traits of crowd workers, have been analyzed in the crowdsourcing literature, aiming to understand the characteristics of workers who submit reliable labels. For example, when examining the effect of age on worker behavior, it has been found that older workers tend to complete more tasks [9]. Sharing the framing/purpose of a labeling task with the crowd workers has been shown to improve their performance [16]. The problem of obtaining labels from experts versus non-experts has been investigated for diverse tasks [2, 17, 8]. The general trend emerging from these works is that experts rarely provide more reliable labels than non-experts. Instead, most of the time both groups provide labels of similar quality. Consistency, which might be affected by training, expertise, or fatigue emerging during a crowdsourcing task, has been proposed as a measure for workers' reliability [18]. Consistency is measured by letting workers label previous documents again and if they consistently assign the same label, it indicates that their labels are generally more reliable. In [19], 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 [20] (crowd workers might overestimate their expertise w.r.t. a topic and therefore try to compensate for it with general knowledge), also contribute to the reliability of workers.

In [21], Calma et al. point out that workers, called "oracles" in their work, can vary in their expertise and be uncertain in their decisions for various reasons. Calma et al. propose that oracles collaborate with each other and with the active machine learning algorithm to achieve better performance [21]. Collaboration is out of scope of our work, since we want to understand first whether and to what extend workers are (un)certain, but we expect that some of the sources of uncertainty mentioned in [21], namely boredom and fatigue, can be traced in the temporal dynamics of crowd workers that we investigate.

Several works attempt to predict the quality of the labels delivered by the workers by analyzing solely behavioral features like annotation time, mouse clicks and scrolling behavior [22, 23, 24, 25]. In [26], Han et al. combine behavior data with a worker's historical data, e.g. the performance over the last 10 submitted crowdsourcing tasks, and

show that predictors trained on such data are more robust against cheating than predictors trained on behavioral features alone. To avoid low-quality labels, Kara et al. propose a new metric to measure worker quality in crowdsourcing settings which takes worker behavior into account [6].

2.2 Annotation Time as a Human Factor

Annotation time is a behavioral feature of human workers and describes the time needed for a worker to assign a label to a document. This feature is widely used to draw conclusions on workers' performance and on label quality, see e.g. [27, 28, 24, 29]. We denote this time as *annotation time* or as *labeling cost*: this second term comes from active machine learning, see e.g. [30], [31], and [32] because the more time a worker needs for the annotation, the higher costs incur if one assumes a limited time that is available for finishing the whole labeling process. In that case higher annotation times imply less labeled documents. Zhu et al. show that workers' behavior over time is indicatory of their reliability [7]: they monitor the time needed to annotate a document and point out that the time curve for "normal" workers sinks rapidly in the beginning and then remains roughly the same in the rest of the annotation task. Zhu et al. consider spikes as indicatory of distractions from the annotation work, and cast doubts on the reliability of the labels thus produced [7]. This is one of many studies that leverage annotation time to discriminate between reliable and unreliable workers.

The analysis of the temporal dynamics of workers' activities is a much rarer subject. In [13], Settles et al. study annotation dynamics in order to optimize active machine learning strategies. They report that after the annotation of only a few documents, the *labeling cost*, defined as the time required to label a document, converged toward a constant value [13]. This is in agreement with [7], who expect that the annotation time per document converges rapidly and does not change thereafter.

Insights on the convergence process itself are even more seldom. An indirect finding is reported by Baldridge et al., who investigate the performance of an active machine

learning strategy when the labels are delivered by a human expert vs a human non-expert [30]. These authors found that predictors trained on labels obtained from non-experts caught up with predictors from expert labels after roughly 6 hours in the annotation process [30]. This finding suggests that convergence of the annotation time is not always fast and smooth. In RQ1, we drill into the temporal dynamics of workers' behavior to shed more understanding on how annotation time changes as a worker sees more and more texts.

2.3 Worker Disagreement in Crowdsourcing

There are two schools of thought on worker disagreement in crowdsourcing. According to the first one, worker disagreement is noise and therefore it should be minimized in datasets as only datasets with low disagreement will be useful for training predictors that generalize well. To minimize worker disagreement, an experimenter would have to provide labeling instructions for crowd workers that cover all possibilities in order to teach workers to label the documents according to the instructions. For example, in the subjective task of sentiment analysis, experimenters could reduce worker disagreement by defining certain rules, e.g. "if a document contains positive and negative sentiment, select 'negative' as the label". In contrast, according to the second interpretation, worker disagreement may be harnessed: "[crowd worker] disagreement is not noise, but signal" [33]. That means the fact, that workers disagree on the label of a document, indicates that this document could be interpreted in multiple ways – it does not necessarily imply that any of the crowd workers is unreliable. Aroyo et al. argue in [33] that worker disagreement reflects the true labels of the documents better because providing instructions that cover all possibilities artificially reduce disagreement, yet the resulting datasets might not result in accurate predictors. Instead, the crowd workers' subjective interpretations of the documents are more realistic and datasets labeled in this way eventually lead to more accurate predictors. We adopt the idea that worker disagreement is a signal in this thesis. More precisely, we interpret the presence of disagreement as an indicator for the difficulty of a

document, i.e. the more workers disagree on a document, the more difficult we consider it to be. This assumption forms the basis for Chapter 5 and Chapter 6.

Regardless of the two different interpretations of worker disagreement, disagreement in crowdsourcing is analyzed in different contexts. For word sense annotations it was found that it is easier to predict high disagreement than lower levels of disagreement [34], which is why we model it as a binary classification task in RQ2 and RQ3. 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 [35]. Others find that training workers reduces disagreement [36] and that some strategies for training workers are more promising [37]. It was shown that high/low Kappa/Krippendorf's alpha values, which both measure worker disagreement, do not necessarily correlate with predictor performance [38]. For example, low worker disagreement could have been artificially achieved by workers preferring one specific label over others due to the experimenters providing explicit labeling instructions that enforce the use of one label in certain situations. These instructions might stem from the idea that worker disagreement is noise and must be minimized as explained above. 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 on Twitter in this thesis, we do not provide sample tweets from the dataset to explain the labels, just a short, general description with imaginary, simple tweets to avoid introducing any bias.

2.4 Document and Tweet Difficulty

Martinez et al. utilize a predictor's certainty to approximate the difficulty of a document [39]. The underlying assumption is that a predictor is less certain about predicting labels for difficult documents. We employ the same idea in this thesis to derive tweet difficulty heuristically. Label difficulty has also been acknowledged and researched in the context

of active machine learning [40] and crowdsourcing [41]. However, Gan et al. [41] focus on modeling the difficulty of labeling tasks in crowdsourcing instead of single documents. Paukkeri et al. [42] propose a method to estimate a document's subjective difficulty for each user separately based on comparing a document's terms with the known vocabulary of an individual.

Sameki et al. model tweet difficulty in the context of crowdsourcing [11] where they devise a system that minimizes the labeling costs for micro-tasks by allocating more budget to difficult, i.e. ambiguous, tweets and less to non-ambiguous ones. The authors argue that more sentiment makes a tweet more difficult to understand. Hence, they formulate the problem of estimating tweet difficulty as a task of distinguishing sarcastic from non-sarcastic tweets. One of the factors that they utilize is worker disagreement - if more individuals agree on a label, it is considered easier. That means they also treat worker disagreement as a signal. An approach that is related in spirit to the idea expressed by Sameki et al. [11] is estimating the difficulty of queries [43]: topic difficulty is approximated by analyzing the performances of existing systems - a lower performance indicates more difficult topics. In our work, we also harness worker disagreement to approximate tweet difficulty - lower worker disagreement is associated with non-ambiguous tweets. While this thesis bears similarities with [11], the objectives differ: in RQ2 we are explicitly interested in analyzing how tweet difficulty affects the reliability of tweets that workers assign, while Sameki et al. employ tweet difficulty as a feature to predict the number of workers that should label a tweet. Furthermore, we combine worker disagreement with two more factors to model tweet difficulty for RQ2. In terms of RQ3, the objective of Sameki et al. is to identify tweets that must be labeled by more workers while our objective is to find the tweets that may be treated differently before being given out for crowdsourcing at all. Therefore, in RQ3 we are the first ones to demonstrate how predictor performance is affected by removing tweets with *Disagreement* compared to allotting more workers to them. Another approach related to this thesis is described in [44] where the authors propose a probabilistic method that takes image difficulty and crowd worker expertise into account to derive a ground truth – the authors show that this idea is more accurate than

majority voting. However, they do not consider that workers learn during a labeling task. In addition, we focus on analyzing how the performance of predictors is affected by tweet difficulty.

We do not adopt any of the proposed methods in text mining to model difficulty, e.g. [45], although tweets are also texts. This is because tweets are too short to extract meaningful grammatical features and sometimes they even do not contain any well-formed sentences at all. Therefore, we model tweet difficulty using the abovementioned heuristics from the crowdsourcing context which correlate intuitively with tweet difficulty.

Chapter 3

Materials and Basic Methods

First, this chapter describes how we acquired the dataset used for analyses in the following chapters. Parts of this work appeared in [46]. In addition, we describe how to compute the pairwise similarity of tweets which is used to answer RQ2 and RQ3.

3.1 Building the Dataset for the Experiments

Experiments with human subjects require a careful design process in order to obtain a reliable dataset for analysis. Figure 3.1 illustrates the different subtasks we performed to produce our initial dataset TRAIN. After designing the annotation experiment, we devised an annotation protocol, implemented our web-based annotation tool and collected a Twitter dataset which is suitable for the labeling task we have chosen. After preprocessing the Twitter dataset and storing it in a database, we recruited volunteers as crowd workers who participated in the actual experiment in a controlled environment. The task given to the workers was to label the tweets according to the hierarchical labeling scheme described in Figure 3.2. We use the resulting labeled dataset for investigating our research questions. The following subsections describe all aforementioned steps in more detail.

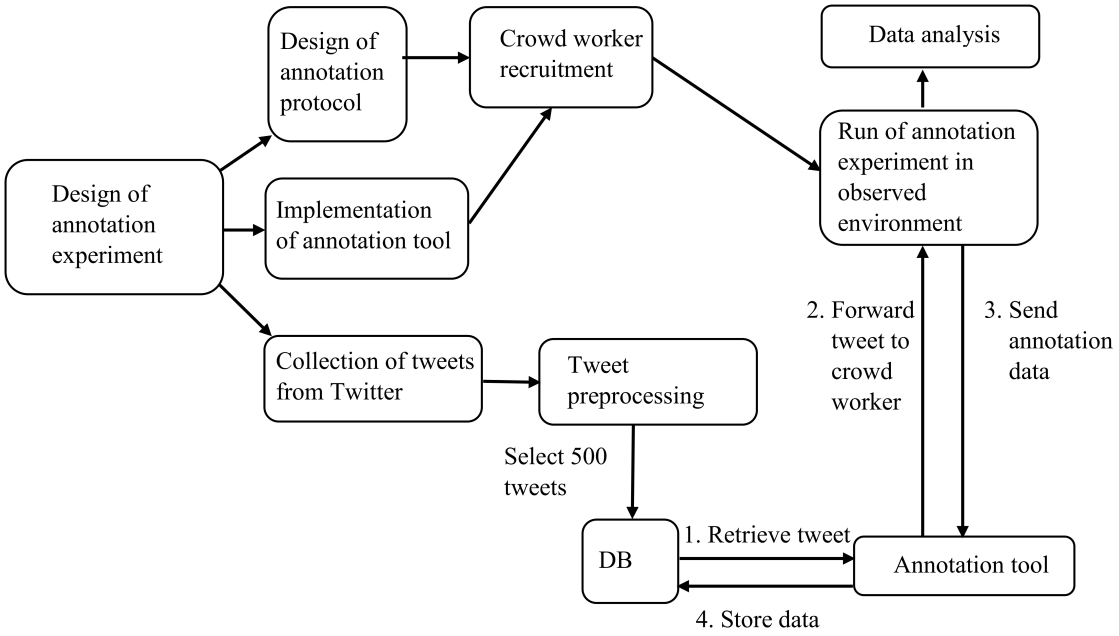


Figure 3.1: Workflow for the annotation experiment and the analysis of the crowd workers' data.

3.1.1 Collecting the Dataset

We collected from Twitter 240k tweets with the Twitter Streaming API on 27 September 2016 during the first public debate (9-10.30pm EST) between Donald Trump and Hillary Clinton using the hashtags #debatenight, #debates2016, and #debates. In the preprocessing phase we kept only unique tweets which did not contain any URLs or attachments. We also selected tweets to contain at least 23 words[1]. These tweets form dataset TRAIN. Choosing tweets with a high number of words increased the probability that a sentiment was expressed in those tweets. Tweets meeting the above preprocessing criteria but containing fewer words were added to dataset C instead. As a result, TRAIN contains 500 tweets, while C comprises 19.5k tweets. In addition, we used a subset of 200 randomly selected tweets from TRAIN to build TRAIN$_S$.

[1]We calibrated this number so that we have a significant number of words in a tweets and also making sure that we have 500 tweets remaining in the dataset after the preprocessing.

3.1.2 Designing the Annotation Experiment

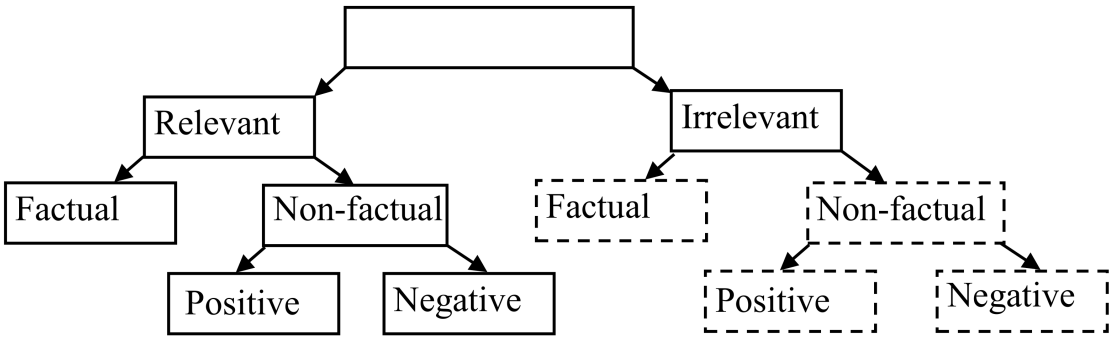


Figure 3.2: Annotation scheme for the hierarchical labeling task. Labels with dashed outline are removed from the dataset. Note that each hierarchy level corresponds to one of the three label sets: *Relevant* vs. *Irrelevant*, *Factual* vs. *Non-factual*, and *Positive* vs. *Negative*.

To study the change of labeling costs over time, we prepared sets of tweets in three different sizes, S (small set of 50 tweets), M (medium-sized set of 150 tweets), and L (large set of 500 tweets) as explained below. With this, we could check whether the number of tweets to be annotated affects labeling costs, e.g. because crowd workers learn what makes a tweet negative (for example) and assign labels faster, or because they get distracted or tired over time.

To build the set S of tweets, the annotation tool chose randomly 50 tweets from TRAIN_S for each crowd worker belonging to group S and 150 tweets from TRAIN_S for crowd workers of group M. The reason for sampling from TRAIN_S instead of TRAIN is that we wanted each tweet to be labeled multiple times. Only workers of group L labeled all tweets of TRAIN. Consequently, sets of tweets to be labeled by crowd workers from S and M may be different but overlapping. Crowd workers from groups S and M labeled tweets in an uninterrupted session of approximately 90 min, while workers from group L performed their labeling tasks in three separate sessions of at most 90 min each. 150, 200, and 150 tweets were labeled in the first, second, and third session respectively. Workers had to take a break between each session for at least 30 min.

The recruitment of crowd workers, see Figure 3.1, middle upper part, was the next step. We recruited crowd workers from two geographic regions, namely from Magdeburg (MD) in Germany and from Sabancı (SU) in Turkey to investigate the generalizability of our results for RQ1. Since it is known that providing crowd workers with different information about a task affects their labeling behavior [47], we prepared an annotation protocol with the same information for all participants to ensure that they start with similar background knowledge. In addition, the annotation experiment was run in a controlled setting, a class room in our case, where one of the designers of the experiment was available at all times to assist the participants if they encountered any problems and to ensure that they did not influence each other by talking.

The annotation tool[2], see Figure 3.1, right lowermost part, chose randomly the tweets to be presented to each crowd worker. One such sample tweet from TRAIN is shown below:

```
Did trump just say there needs to be law and order
immediately after saying that he feels justified not
paying his workers??  \#Debates
```

Figure 3.3 displays a screenshot of the web-based annotation tool we implemented. Our annotation tool simulates a crowdsourcing environment where users log in to perform specific labeling task. Crowd workers were given the task of determining the sentiment expressed in each tweet presented to them.

To enforce the labeling scheme of Figure 3.2 and prevent contradictory label assignments (e.g. labeling a tweet as *Factual* and *Negative*), our implemented annotation tool presented first the pair of labels *Relevant* and *Irrelevant* as shown in Figure 3.3. Once a crowd worker chose a label, *Factual* and *Non-factual* appeared. Similarly, *Positive* and *Negative* were displayed if and only if crowd workers decided for *Non-factual* as *Factual* represents *Neutral* tweets[3]. For each of the labels to be assigned, crowd workers had to assess the confidence in their own label choices by selecting as a label either *High* or

[2] `https://github.com/fensta/annotationtool`

[3] We use *Factual* and *Neutral* interchangeably throughout this thesis.

Low. This way, each crowd worker assigned either two or three annotation and confidence labels to a tweet. Besides the labels, the annotation tool stored for each label set the times needed for picking a label (which we call the annotation time) and the time needed to select a confidence label (called confidence time). Additionally, we stored the order in which a crowd worker labeled her tweets. Thus, we can easily identify the labels of the i^{th} tweet for a given worker. To display the next tweet to be labeled, the tool first randomly picked a tweet from the backend (MongoDB in our case) and displayed it to the crowd worker in the web frontend. Once a crowd worker finished labeling the given tweet, all annotation and confidence labels as well as annotation and confidence times were stored in the backend and the next tweet to be displayed was picked randomly again. The tool stopped once the number of tweets specified by the crowd worker group had been labeled.

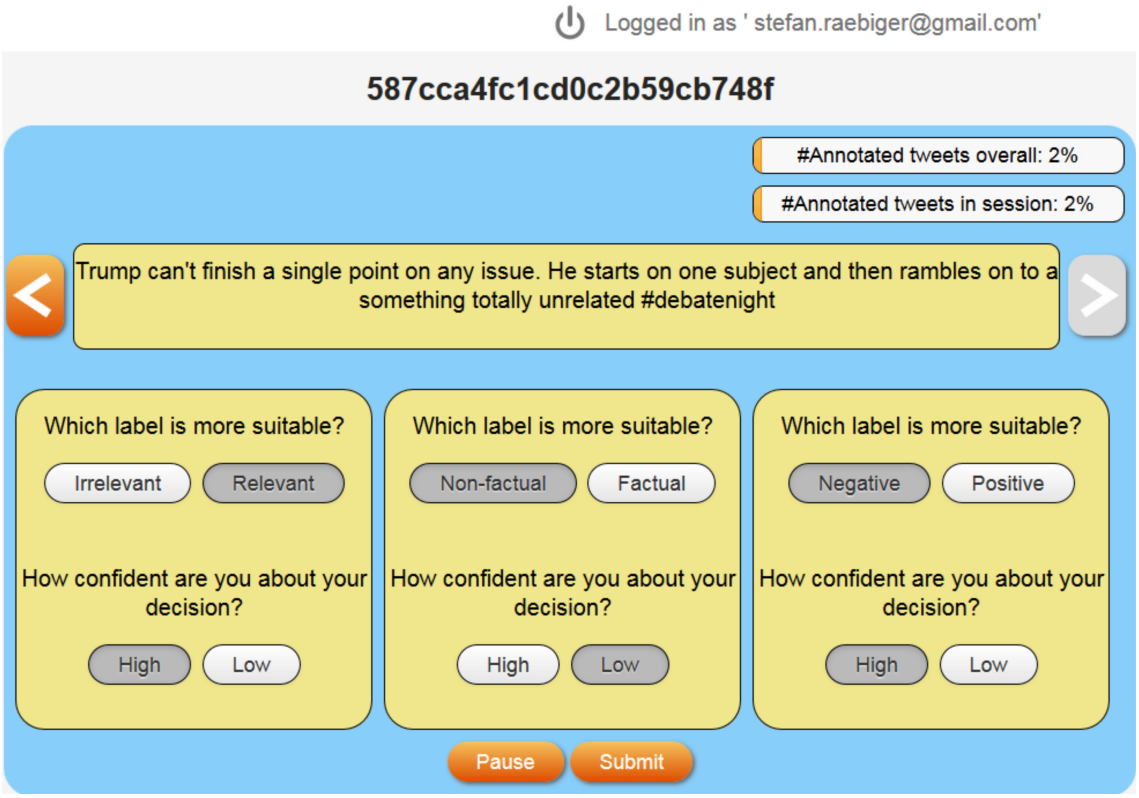


Figure 3.3: Screenshot of the annotation tool displaying all three sets of labels to be assigned. The number in bold on top is the database ID of the tweet.

Institution	Group			Total	Labeled tweets
	S	M	L		
SU	13	9	3	25	3500
MD	10	8	1	19	2200

Table 3.1: Worker distribution and total number of labeled tweets per institution. Group S labeled 50 tweets, group M labeled 150 tweets, and group L labeled 500 tweets.

3.1.3 Labeling the Dataset

In total, 44 students participated in our annotation experiments in MD and SU. The crowd workers in both institutions were from different countries, with a similar gender distribution (60% male), had heterogeneous working experiences, were of similar age (20-30 years old), bachelor or graduate students, but all with a background in computer science. The main difference between both institutions was the way workers were recruited. The annotation experiment was carried out as part of a lecture in SU, while it was conducted with volunteering students in their spare time in MD. Thus, the motivation among workers of MD might have been higher than in SU. The experiment was run over the course of three weeks in MD, as opposed to SU where it was performed within one lecture. The worker distributions of MD and SU are shown in Table 3.1 indicating that we had a similar number of participants per worker group. Two workers from MD participated in group S and M with a break of more than one week in between both labeling sessions. Furthermore, they labeled different tweets in each session. Otherwise the groups S and M were completely disjoint.

3.1.4 Analyzing the Dataset

In this section we explore the basic properties of TRAIN in MD and SU. Specifically, we focus on the distribution of confidence and sentiment labels as well as the time required for crowd workers to assign sentiment labels.

Distribution of Sentiment and Confidence Labels

We first report the distributions of the sentiment and confidence labels in TRAIN for MD and SU. These distributions are shown in Figure 3.4 separately for worker groups S, M, and L respectively. It turns out that the trends are similar in MD and SU which becomes more obvious in Figure 3.5 when group L is discarded due to the few number of participants: most tweets are deemed *Relevant* ($> 30\%$) and *Negative* ($> 20\%$). However, there are subtle differences in the sentiment label distributions, namely group S of SU assigned *Relevant* more frequently than their counterparts in MD. In terms of confidence labels, participants of group S in SU were more confident in their label choices than their counterparts in MD. Nevertheless, the differences are minor and we interpret that as an indicator that the crowd workers labeled the tweets faithfully.

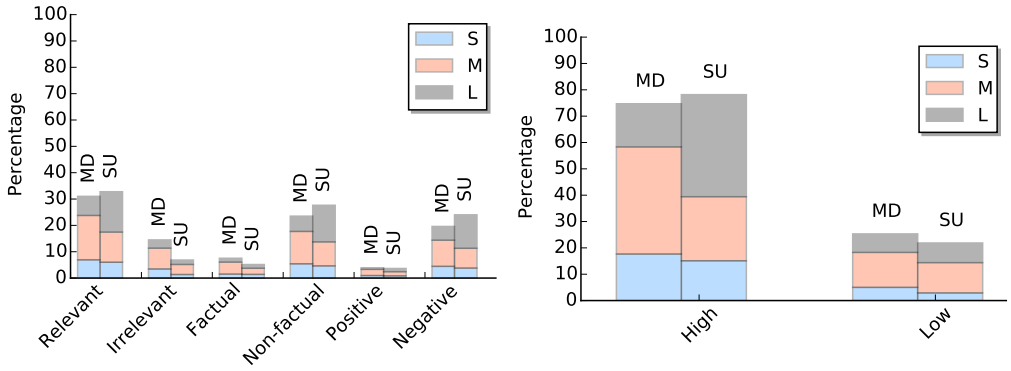


Figure 3.4: Distribution of sentiment and confidence labels for worker groups S, M, and L. Left: label distribution. Right: confidence label distribution.

Median Annotation Times

Annotation times represent the costs for labeling tweets. The longer the annotation process takes for a single tweet, the more expensive the acquisition of the label gets as crowd workers need to be compensated appropriately. We report the labeling costs for each label separately. For aggregating the costs, we use medians instead of averages because the former is more robust towards outliers which occur at times in TRAIN. Therefore, we use median annotation times throughout the thesis when having to aggregate annotation

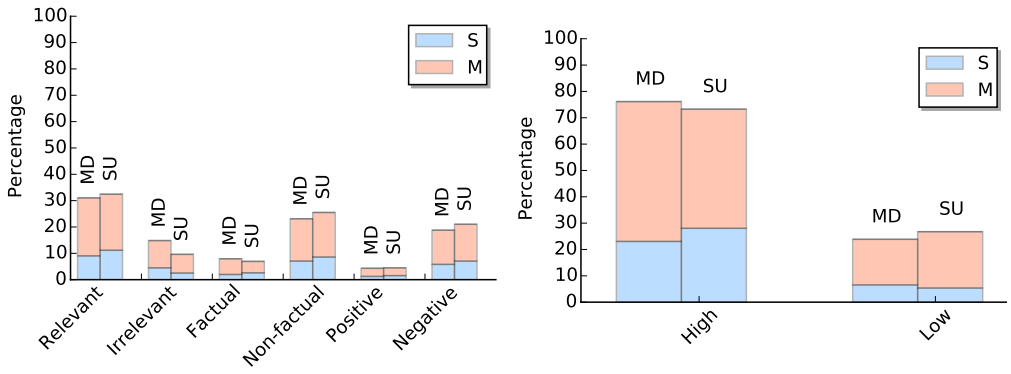


Figure 3.5: Distribution of sentiment and confidence labels for worker groups S and M. Left: label distribution. Right: confidence label distribution.

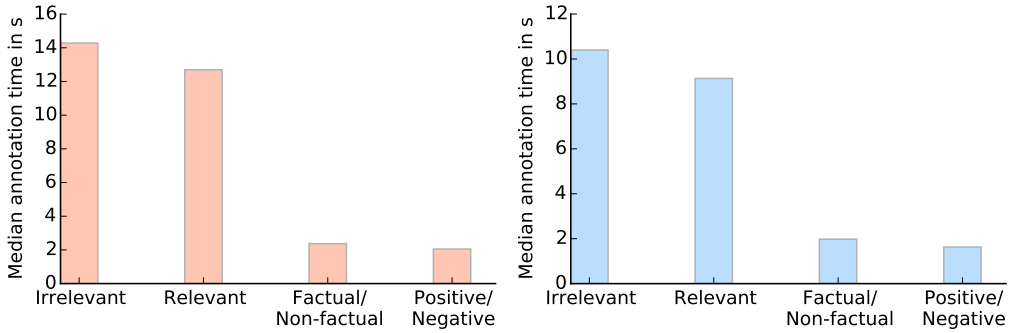


Figure 3.6: Median labeling costs per label. Left: MD. Right: SU.

times. In Figure 3.6, we visualize these median annotation times separately for each label in MD and SU. Results show the same tendency in both SU and MD where most of the annotation time (9 to 14 s) is spent on choosing a label from the first set of labels, while selecting appropriate labels for the other label sets takes about 2 s each. This behavior is expected since one first needs to read and understand a tweet before assigning labels. The only difference between SU and MD is that workers in MD need approximately 4 s more time to assign a label for the first set of labels.

3.1.5 Cleaning the Dataset

For all analyses throughout this thesis, we only consider ”cleaned” datasets, meaning whenever a tweet was assigned the label *Irrelevant* by a crowd worker, only the anno-

tation time for the first label set is considered as labeling cost and all other labels and corresponding annotation times assigned to this tweet by that worker are discarded. During the annotation experiment we did not want our labeling hierarchy to be too skewed as this could have biased the workers' labeling behavior over time. For example, workers could have been more likely to assign *Irrelevant* once they noticed that no other labels had to be assigned in this case. Hence, we opted for letting workers also assign the remaining label sets. However, in practice we would not proceed with labeling such tweets beyond the first label because we are only interested in tweet sentiment of relevant tweets, which is why we focus on the cleaned datasets.

3.2 Methods for Comparing the Similarity of Short Documents

In our experiments addressing RQ2 and RQ3, we employ a kNN predictor. Therefore we must establish a similarity between any two tweets $t1$ and $t2$. Since tweets may exhibit different lengths, we normalize this similarity by the longer tweet to avoid any influence of the text length on the similarity. Therefore, this normalized similarity yields values between zero (tweet texts are disjoint) and one (identical tweets). We refer to this normalized similarity as $NormSim$ and it is computed between $t1$ and $t2$ as:

$$NormSim(t1, t2) = sim(w1, w2)/max(|w1|, |w2|) \tag{3.1}$$

where $w1$ and $w2$ represent the words in the tweets $t1$ and $t2$ and $sim(w1, w2)$ computes the number of shared words between $t1$ and $t2$ according to a similarity metric. In this thesis, we utilize as metrics:

1. Longest common subsequence
2. Longest common substring
3. Edit distance

These three metrics are typically defined on character-level, i.e. they compute the similarity between two single words by comparing them character by character. Since we deal with tweets containing multiple words, we apply the metrics on word-level instead. Edit distance between two strings counts how many characters in one string need to be changed to transform it into the other string on the character-level. However, when focusing on the word-level, edit distance counts how many words in tweet $t1$ must be replaced s.t. it results in tweet $t2$. Similarly, longest common subsequence counts how many characters in both words are in the same relative, but not necessarily contiguous, order in terms of character-level. Extending this to idea to word-level means this metric now counts the words in $t1$ and $t2$ that are in the same relative, but not necessarily contiguous, order. Last, but not least, longest common substring counts how many contiguous characters both words share on the character-level. That means this metric counts on the word-level the number of words that are contiguously shared among $t1$ and $t2$.

For $NormSim$ to yield values between zero and one, the term $sim(w1, w2)$ needs to be inversed when using edit distance because large values indicate that $t1$ and $t2$ are different as opposed to being similar. Thus, when using edit distance, we use $1 - sim(w1, w2)$ instead of $sim(w1, w2)$ in Equation 3.1.

Chapter 4

The Annotation Behavior of Crowd Workers over Time

In this chapter we investigate RQ1, i.e. how the reliability of labels assigned by crowd workers develops over time. To do so, we first analyze how workers learn during a labeling task. Specifically, we focus on the dynamics of annotation times, i.e. the times needed by crowd workers to assign labels. With these identified patterns in mind, we investigate how these affect the label reliability of crowd workers. First, we describe our assumptions about how we expect crowd workers to learn in labeling tasks and we formulate specific, refined research questions in Section 4.1. Section 4.2 describes the methods used for answering these questions and Section 4.3 reports our results. Section 4.4 discusses applications of these results and possible avenues for future research. Parts of this chapter appeared in [46].

4.1 Introduction

Crowdsourcing is a widely used means to label large-scale datasets, but the labels thus produced by the human crowd workers often lack the desired quality: the studies of [4, 5, 6] attribute errors and inconsistencies to spammers, inexperienced workers and workers without adequate motivation. How do workers behave when they assign labels though,

assuming that they are neither spammers nor unmotivated? The temporal dynamics of the text labeling task, namely the process of *learning* what makes a short piece of text positive or negative with respect to sentiment have been rarely the subject of investigation thus far.

To the best of our knowledge, the connection between labeling costs/annotation time (total time needed to label one document) and the learning phase as well as label reliability and learning phase have not been analyzed yet. Our aim is to investigate these issues in a hierarchical crowdsourcing tweet labeling task. Our main assumption is that, for a given complex labeling task, workers need to learn a conceptual model, to which we also refer as *worker's model*. We assume that such a model includes background knowledge about the task. For example, if the task is about assigning sentiment labels to tweets involving "Dark Souls III", workers might initially know little about the topic. However, after looking at some tweets, they could learn that it is a challenging role-playing computer game in a Dystopian world. Learning a model in this context means that crowd workers refine their initial conceptual models over the first few encountered documents. While doing so, the labeling costs, which could be identified by a significant drop in annotation times, are expected to decrease. We refer to this phase as *worker's learning phase*. Then, after some time these labeling costs are expected to converge to a roughly constant level, to which we refer as *worker's exploitation phase*. In short, the exploitation phase begins directly after the learning phase.

We expect our assumption to hold if and only if the labeling task is sufficiently challenging, meaning that crowd workers do not have a perfect conceptual model at hand in the beginning. Otherwise they can solve a task without requiring any additional knowledge, therefore no learning phase would occur. For example, if the labeling task would be to identify the picture that contains a human from a set of three pictures, workers would easily solve the problem because they intuitively know the characteristics that describe humans.

In light of the above discussion, we investigate the following research questions:

- **RQ1.1**. Which factors affect the duration of the labeling process?

- **RQ1.2**. How do institution and worker group affect the labeling costs?
- **RQ1.3**. Does the variance of labeling costs reduce toward the end of the labeling task?
- **RQ1.4**. How reliable are the labels obtained from the learning phase?

4.2 Methods for Analysis

Due to the low number of participants in group L we ignore all tweets labeled by these crowd workers in our analysis. In the following, we first discuss the design of our analysis that is common to *all* RQs (RQ1.1- RQ1.4), and then we discuss how we studied each RQ separately.

4.2.1 Elements of the Data Analysis Common to all Research Questions

Modeling Labeling Costs of Individual Tweets For a given crowd worker, we approximate her labeling costs for a single tweet as the total time needed to assign all tweet labels.

From Individual to Aggregated Labeling Costs In RQ1.1, RQ1.2, and RQ1.3, we are interested in labeling costs over time for worker groups and institutions. To aggregate labeling costs of single workers, we consider medians instead of averages throughout this chapter as the former are more robust to outliers as described in Section 3.1.4. The details are given in the respective sections.

Significance tests Whenever we perform significance tests, we accompany them with visual analyses. We always employ the two-tailed Wilcoxon rank sum test (cf. Appendix A.1) with significance level $\alpha = 0.05$ since our labeling costs are not normally distributed. Only in RQ1.3 we use ANOVA (cf. Appendix A.2) after log-normal transform-

ing the labeling costs. The two-tailed variant of Wilcoxon rank sum test is appropriate in our case because we already know the direction of the relationship from visual inspection and our ultimate goal is to tell if differences between labeling costs are significant, but not if they are significantly larger or smaller. Unless otherwise stated, we compare the first i and the last i labeling costs of workers because ultimately, we want to determine if there are differences between the initial and final labeling costs. Since our annotation tool randomly selected the tweets to be labeled by a crowd worker, our collected data could suffer from order effects. For example, a crowd worker might have labeled most of the ambiguous tweets at the beginning of the annotation session while non-ambiguous tweets were labeled toward the end of the session. Assuming that annotating non-ambiguous tweets is in general faster than labeling ambiguous ones, crowd workers' labeling behavior over time could be easily misrepresented in this case. We account for such potential order effects in our user data by shuffling a worker's first i annotation times and her last i annotation times separately, thus turning the ordered annotation times into sets. In this scenario the unpaired Wilcoxon rank sum test is applicable. Furthermore, we verified that the positions of the tweets, as they were seen by individual crowd workers, were random, i.e. they occurred with equal probability at any position during annotation sessions. Therefore, our statistical results do not suffer from any bias introduced due to labeling ambiguous tweets toward the beginning and most non-ambiguous ones toward the end of an annotation session, assuming that non-ambiguous tweets are labeled faster than ambiguous ones.

Cleaning the data As described in Section 3.1.5, we discard sentiment labels of *Irrelevant* tweets from our analysis. One might argue that this more skewed labeling scheme could bias our data in favor of significant results, but we performed significance tests on the raw and cleaned version of TRAIN and obtained identical results. We omit the raw results to facilitate readability.

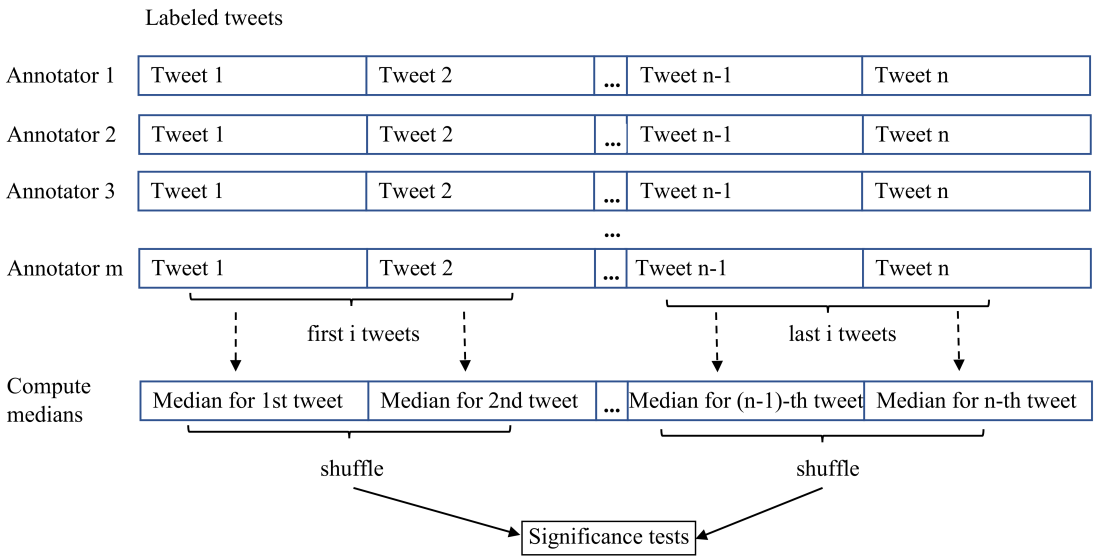


Figure 4.1: Overview of how the median labeling costs for the first i and last i tweets of workers in a specific group are computed, which then serve as input for significance tests.

4.2.2 Factors Affecting the Length of the Labeling Process

As mentioned in Section 4.1, RQ1.1 analyzes which factors affect the duration of the labeling process. We quantify the duration of a worker's learning phase and exploitation phase by the number of tweets she labeled in the respective phase. First, we perform a significance test and afterwards, we complement these results with a visual analysis of workers' labeling behavior. In the significance tests, we compare the first i labeling costs of workers with their last i labeling costs graphically by plotting the resulting p-values when varying i. We expect to obtain significant differences in the labeling costs up to a certain i. After reaching this value, which indicates that a worker finished learning her worker's model, the differences should remain non-significant for the remaining tweets. That means i marks the end of a worker's learning phase and the beginning of her exploitation phase. Since we are interested in observing labeling costs for worker groups and institutions, we aggregate labeling costs by using the median cost for the i^{th} tweet labeled by workers of the respective group in an institution. For comparing the first i with the last i labeling costs in group G, where $G \in \{\text{S}, \text{M}\}$, within an institution U, where

$U \in \{\text{SU, MD}\}$, we create for G two sequences representing the first i and last i median labeling costs respectively. Each sequence comprises exactly i values. This process is illustrated in Figure 4.1. Assuming that there are m crowd workers in G, we obtain the values for the first i (last i) median labeling costs as follows: for the i^{th} labeled tweet, where $i \leq$ first i ($i \geq$ last i), of each worker belonging to G, we select the median from the labeling costs of all m workers and add it to the list for the first i (last i) median labeling costs. We then convert both sequences into sets by shuffling their values to account for order effects. Our corresponding null hypothesis to test is:

1. H_0: in G of U, there is no difference between the set of the first i median labeling costs and the set of the last i median labeling costs, where $U \in \{\text{SU, MD}\}$ and $G \in \{\text{S, M}\}$.

We assume that the learning phase in a worker group is completed once a p-value for a given i exceeds the significance level. However, since some p-values are close to the significance level and could thus also be just outliers, we focus on the overall trends of the p-values that we obtain from varying i.

To complement the significance tests, assuming that there are n tweets, we analyze how the labeling costs for the first i and for the remaining $n - i$ tweets develop. Specifically, we plot for each G the median labeling costs for the i^{th} tweet. Then we fit a polynomial of degree three to the first i tweets within a worker's learning phase and another polynomial of degree three to the remaining $n - i$ tweets. The parameter i thus creates two intervals. For both polynomials we compute and plot the second derivatives, which represents acceleration. Since we expect a worker's learning phase to be completed after seeing the first i tweets, the slope of the corresponding acceleration should be negative. Similarly, the slope of the acceleration should be roughly zero in the exploitation phase for the remaining $n - i$ tweets. We test our hypothesis by varying the parameter i and creating multiple plots.

4.2.3 The Effect of *Worker Group* and *Institution* on Labeling Costs

As mentioned in Section 4.1, RQ1.2 investigates if any of the factors worker group (either S or M) and institution (either MD or SU) affects the workers' labeling costs over time. Therefore, we perform a graphical and statistical analysis. For visualizing the labeling costs, we display per institution for each worker her median labeling costs, but the workers are ordered according to their worker groups. In the statistical analysis, we examine if there are any differences in the temporal learning behavior between the same groups of MD and SU. We also test if there are differences in the median labeling costs of S and M within an institution. In both cases, we use the first i and last i labeling costs per crowd worker. Each time, we compare the first i labeling costs with each other and repeat the same procedure for the last i labeling costs separately. We use the method described in Section 4.2.2 to determine median labeling costs for the i^{th} tweet that are converted into sets. Our null hypotheses are:

- H_1: in G, the set of the first i median labeling costs in SU is identical with the set of the first i median labeling costs in MD, where $G \in \{\text{S,M}\}$.
- H_2: in G, the set of the last i median labeling costs in SU is identical with the set of the last i median labeling costs in MD, where $G \in \{\text{S,M}\}$.
- H_3: in U, the set of the first i median labeling costs in S is identical with the set of the first i median labeling costs in M, where $U \in \{\text{SU, MD}\}$.
- H_4: in U, the set of the last i median labeling costs in S are identical with the set of the last i median labeling costs in M, where $U \in \{\text{SU, MD}\}$.

4.2.4 Development of the Variance of Labeling Costs over Time

In RQ1.3, as mentioned in Section 4.1, we investigate for each institution if the consensus of workers in terms of labeling costs is affected by their learning phase. One way to express consensus is as variability, specifically as between-subjects and within-subjects

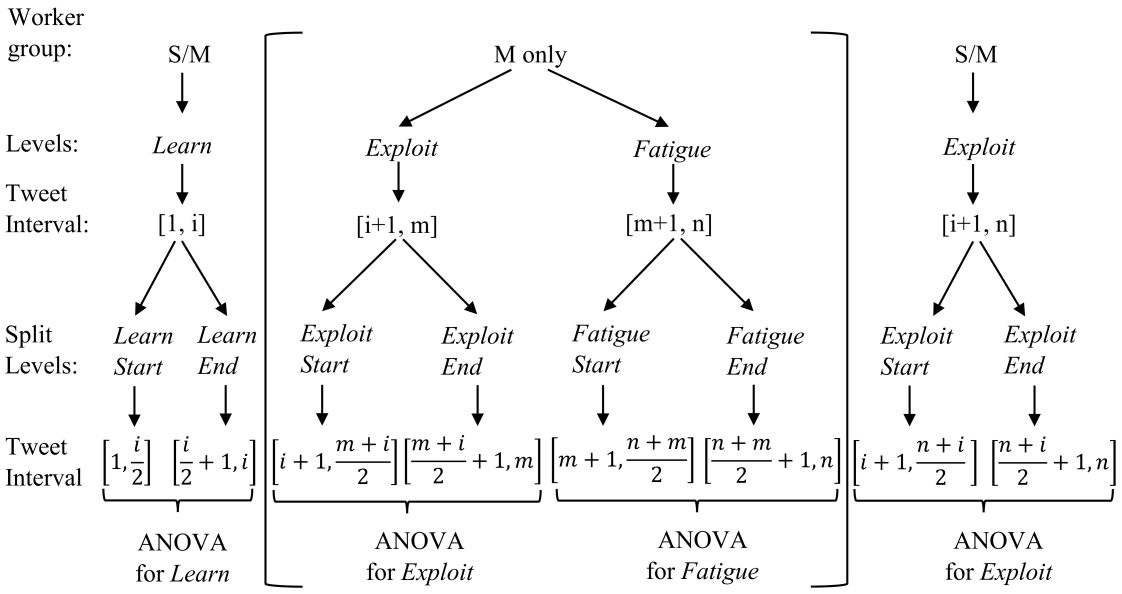


Figure 4.2: Schematic representation of our ANOVA. We assume a worker labeled n tweets in total. Depending on her worker group, a different analysis is performed: for S, the levels *Learn* and *Exploit* are analyzed, while for M two cases are distinguished: using (a) the same levels as in S, and (b) introducing an extra level, *Fatigue*. The tweets falling into the respective intervals of a level are used in ANOVA. For example, for $Learn$, the worker's first i labeled tweets are used. Each level is then split into two sublevels and the intervals are halved correspondingly before performing ANOVA. The parameter i is determined in RQ1.1 and we set m to a reasonable value.

variability. Between-subjects variability here describes the variation of the median annotation times over all workers between the different levels (e.g. "Learn Start" vs. "Learn End" as shown in Figure 4.2). On the other hand, within-subjects variability represents the variability of the median annotation times (of all workers) within a level (e.g. "Learn Start"). To measure both kinds of variability, we essentially perform ANOVA for each institution because the resulting F statistic is computed as $F = \frac{\text{between-subjects variability}}{\text{within-subjects variability}}$. Thus, we only compute and report numerator and denominator. To determine a value for both variabilities once in the learning phase - *Learn* - and once in the exploitation phase - *Exploit* - we perform ANOVA according to the scheme depicted in Figure 4.2. Assuming

that a worker labeled n tweets in total, she labeled the first i tweets in her learning phase. Thus, these i tweets are considered in the level *Learn*, while all remaining $n - i$ tweets belong to *Exploit*. The parameter i is set according to the results of RQ1.1. The dependent variable in ANOVA is the median labeling cost, while the independent one is the position at which a tweet was labeled by a worker. For each level, we choose per worker her median labeling cost in the corresponding interval. Since we want to obtain between-subjects and within-subjects variabilities for *Learn* and *Exploit* separately, we split both intervals into sublevels as outlined in Figure 4.2, e.g. *Learn* is divided into *Learn Start* and *Learn End*. In both cases, we split the intervals in the middle. For group M we also want to test if workers suffered from fatigue towards the end, hence we split *Exploit* into *Exploit* and *Fatigue*. Instead of halving the intervals, we assign $m - i$ tweets to $Exploit$ and the remaining $n - m$ tweets to *Fatigue*. We set the parameter m relatively close to i since we expect fatigue to begin soon after group S finished their tweets. Then we perform ANOVA separately for *Learn*, *Exploit* for groups S and M and if the worker belongs to M, we additionally perform an ANOVA for *Fatigue* after splitting *Exploit* up. Since ANOVA assumes the data to be normally distributed, we log-normal transform our labeling costs.

We expect the consensus to be initially lower than toward the end since workers are still refining their conceptual models. For this hypothesis to hold, between-subjects variability must decrease after *Learn*, while within-subjects variability should remain largely unaffected as individuals label differently. If workers fatigue, we expect between-subjects variability in *Fatigue* to be higher than in *Exploit* but lower than in *Learn* because only a few workers might be exhausted which should be reflected in an increased between-subjects variability.

4.2.5 Label Reliability over Time

RQ1.4, as mentioned in Section 4.1, examines if labels obtained during a crowd worker's learning phase are as reliable as those collected later in her exploitation phase. We perform an experiment to measure the effect of the learning phase on a worker's labeling reliability by casting our problem as a hierarchical classification task. Given n tweets per

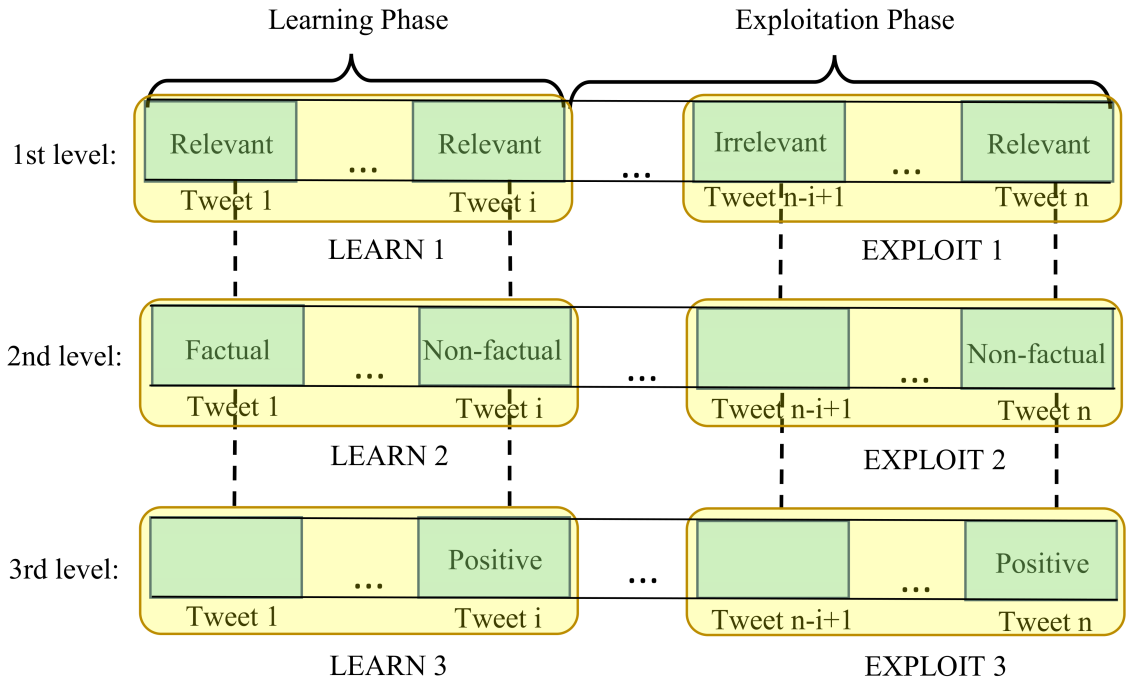


Figure 4.3: Schematic representation of hierarchical classification task. Two predictors are trained per hierarchy level (=label set) for a crowd worker who labeled n tweets. Each predictor is trained on i tweets (marked by yellow) - either the i tweets from the worker's learning phase or her last i labeled tweets. Dashed lines indicate the labels of a tweet on the next lower hierarchy level. In the cleaned dataset, we discarded all further labels if a tweet was assigned *Irrelevant*.

worker, our goal is to predict all her sentiment labels. We do not train predictors across workers since they develop their own subjective conceptual models. For each hierarchy level we build two k-nearest neighbor (kNN) predictors per worker, one, called $LEARN$, trained on all i tweets encountered during a crowd worker's learning phase and one, called $EXPLOIT$, trained on the last i labeled tweets by this worker. The parameter i is chosen according to Section 4.2.2, s.t. a worker's learning phase is completed after she saw the first i tweets. This scheme is illustrated in Figure 4.3. We estimate with both predictors all labels of the remaining tweets. However, in $EXPLOIT$, we also discard all tweets from this worker's learning phase and predict only the remaining unknown tweets, i.e. if that worker labeled n tweets, $LEARN$ predicts the labels of $n - i$ tweets and $EXPLOIT$

predicts the labels of $n - 2i$ tweets. We build the following predictors, where numbers at the end of acronyms describe to which label set/level a predictor is applied.

Predictor 1: core learning algorithm: kNN; acronym: LEARN 1

Predictor 2: core learning algorithm: kNN; acronym: LEARN 2

Predictor 3: core learning algorithm: kNN; acronym: LEARN 3

Predictor 4: core learning algorithm: kNN; acronym: EXPLOIT 1

Predictor 5: core learning algorithm: kNN; acronym: EXPLOIT 2

Predictor 6: core learning algorithm: kNN; acronym: EXPLOIT 3

kNN is motivated by the idea that a worker assigns to a tweet the same label she assigned to earlier seen, similar tweets. To measure the similarity of two tweets $t1$ and $t2$, we compute the edit distance[1] according to Equation 3.1 in Section 3.2. We note that finding the best pairwise similarity measure for tweets is beyond the scope of this thesis. But our chosen measure could be improved, e.g. by incorporating word sentiment of the tweets in addition to edit distance. For example, if both tweets contain three negative words, they become automatically more similar. However, one reason for choosing the three metrics mentioned above is the fact that quotes are common in (sarcastic) tweets [11] and the US presidential candidate debate was controversial, hence the chances for encountering sarcastic tweets are high. Therefore, our selected metrics might be more suitable than expected at first glance.

We measure the performance of the trained predictors in terms of hierarchical F1-score [48] as this is the recommended performance metric for hierarchical classification tasks [49]. By varying k, the number of neighbors to be considered to predict unknown labels in kNN, we obtain multiple F1-scores that we plot.

[1]We also tested longest common subsequence and longest common substring as alternative similarity measures, but the results remained the same.

4.3 Results of Analysis

In this section we report the results obtained for RQ1.1 - RQ1.4 using the methods described in Section 4.2.

4.3.1 Factors Affecting the Length of the Labeling Process

The resulting p-values for H_0 are shown in Figure 4.4 for worker group S and in Figure 4.5 for group M when varying i.

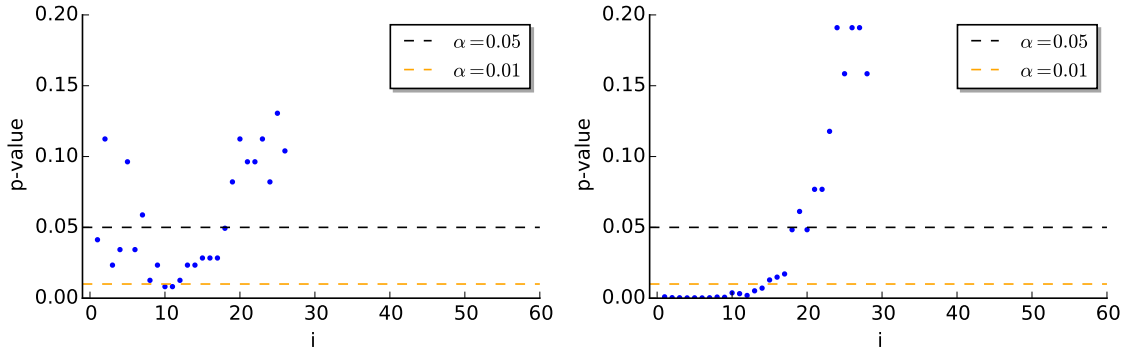


Figure 4.4: p-values when comparing the first i median annotation times with the last i times in group S of both institutions. Left: MD. Right: SU. Missing p-values in both plots for $k > 28$ are > 0.2 and hence not displayed.

It turns out that workers of group S require about 20 tweets for learning their concepts in MD and SU, while workers of group M need around 40 tweets in SU and in case of MD more than 50. Thus, our result indicates that the worker group determines the duration of the learning phase. Therefore, merging the groups to obtain workers' labeling behavior over time for entire institutions would not be meaningful.

The accelerations of the learning curves in MD are shown in Figure 4.6 for group S and in Figure 4.7 for group M. In group S, learning continues after having seen more than 16 tweets, but it is completed before labeling 25 tweets. Since we computed the accelerations for varying i, which represents the number of tweets in the learning phase, we observe that the learning phase for workers of S is completed after seeing around 20

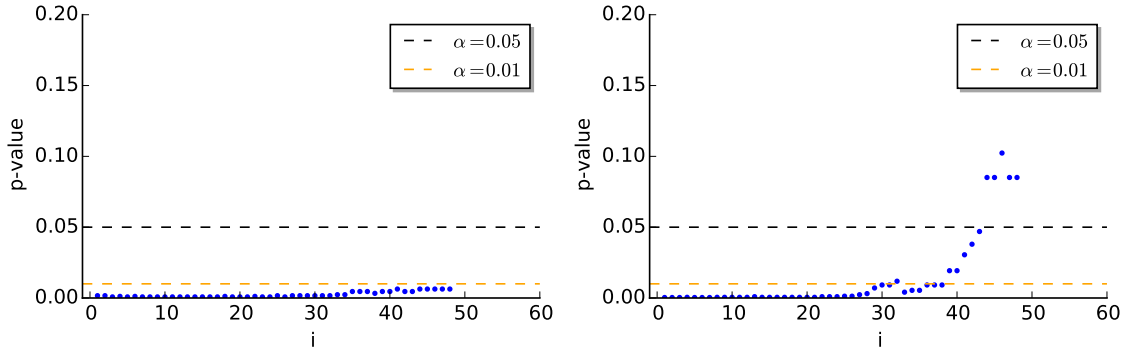


Figure 4.5: p-values when comparing the first i median annotation times with the last i times in group M of both institutions. Left: MD. Right: SU. In neither plot are there any missing p-values.

tweets. In group M of MD, the learning phase continues beyond 30 tweets, but finishes before seeing 41 tweets. According to our plots, the learning phase for workers of M is completed after about 40 tweets. These plots confirm the overall trend indicated by the p-values in Figure 4.4 and Figure 4.5, namely that workers of group S learn faster than those of group M. At the same time, the plots also suggest that the learning phase in M is completed quicker than indicated by the p-values, namely after seeing around 40 tweets. Furthermore, they confirm that the learning phase for workers of group S takes about 20 tweets. The same observations also hold for SU, for which group S is depicted in Figure 4.8 and group M in Figure 4.9.

In RQ1.1 we analyzed which factors affect the duration of the learning process. Summarizing our results on RQ1.1, we found that there are at least two phases with distinct labeling costs, namely a learning phase containing the first 20 (40) tweets and an exploitation phase consisting of 30 (110) tweets in worker group S (M). Thus, the number of tweets to be labeled in an annotation session affects the length of the learning phase. Under RQ1.1, we found that labeling costs change, because workers learn their conceptual model at the beginning and then annotate faster. To investigate RQ1.2, we therefore have to control for this change. To this end, we study the influence of the institution and of the worker group on the (i) first i tweets and separately (ii) on the last i tweets.

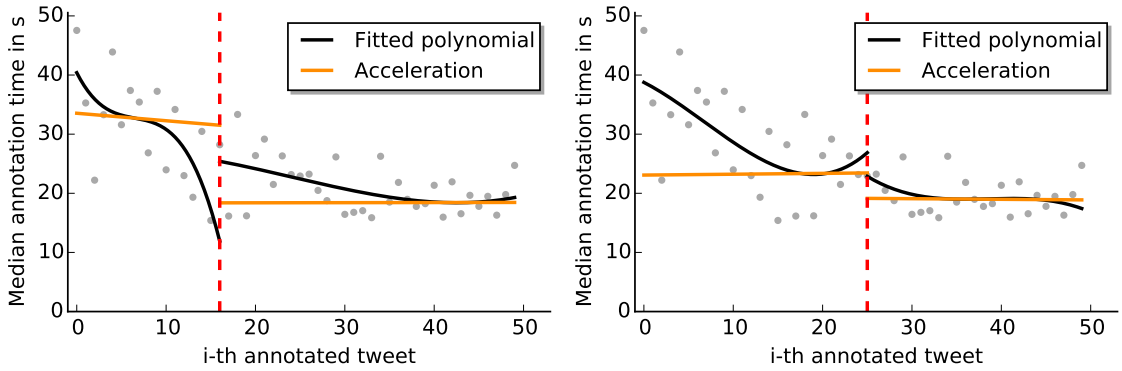


Figure 4.6: Fitted polynomials of degree three and their accelerations for MD (S). Left: the interval boundary (red dashed line) is at $i = 16$ and the change in acceleration in the first interval is negative, so learning is still ongoing. Right: the interval boundary (red dashed line) is at $i = 25$ and the change in acceleration in the first interval is practically zero, so learning is completed.

4.3.2 On *Worker Group* and *Institution* Affecting Labeling Costs

The median labeling costs plotted in Figure 4.10 illustrate that there seems to be a connection between worker groups and median labeling costs in that the costs tend to be higher for workers of group S than for subjects of M.

The p-values are visualized for H_1, i.e. comparing the first i median labeling costs, in Figure 4.11 and in Figure 4.12 for H_2, where only the last i median labeling costs are considered. We find for H_1 that there are some significant differences when comparing group S of MD with group S of SU and the same holds for group M. Interestingly, according to RQ1, these intervals of significant differences (in S: between tweets 10-25, in M: between tweets 30-48) coincide roughly with the transition of workers from their learning phase to their exploitation phase. One possible explanation for the significant differences in these particular intervals might be that workers learn differently, i.e. some learn faster than others, but a more detailed analysis is necessary. For H_2 we obtain non-significant results for group S, but for group M, we obtain significant differences in the labeling costs between MD and SU. From Figure 4.10 we can deduce that workers of SU labeled faster than their counterparts in MD.

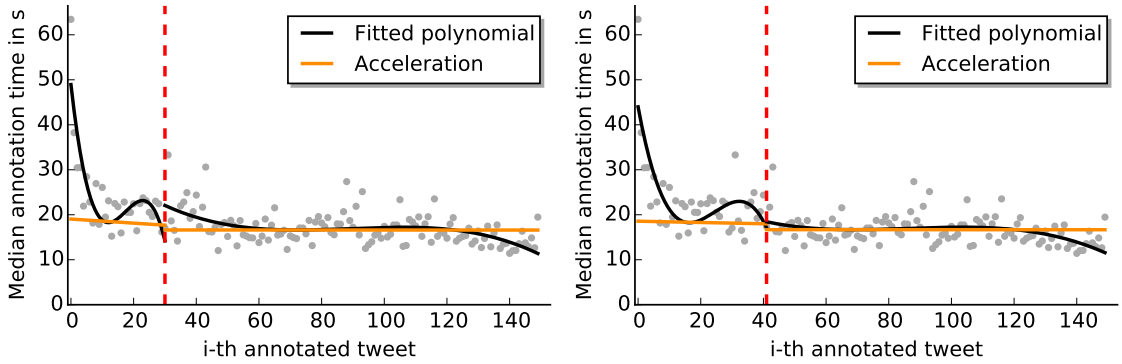


Figure 4.7: Fitted polynomials of degree three and their accelerations for MD (M). Left: the interval boundary (red dashed line) is at $i = 30$ and the change in acceleration in the first interval is negative, so learning is still ongoing. Right: the interval boundary (red dashed line) is at $i = 41$ and the change in acceleration in the first interval is practically zero, so learning is completed.

The resulting p-values for H_3, i.e. comparing the first i median labeling costs between groups S and M in the same institution, are depicted in Figure 4.13 and for H_4, i.e. comparing the last i median labeling costs between groups S and M in the same institution, they are shown in Figure 4.14. For H_3 we obtain non-significant differences for groups S and M, again the p-values are closest to the significance level during the transition of workers from their learning to their exploitation phase. For H_4, we observe significant differences between groups S and M in SU, but rarely in MD, although there are many p-values close to the significance level threshold. In this case it would be necessary to have more data available to see if the p-values decrease.

In RQ1.2 we analyzed how institutions and worker groups affect the labeling costs. Overall, from a visual analysis our results indicate that the labeling costs per tweet are lower for workers of group M, independent of any institution. However, from a statistical point of view the results are inconclusive. Nevertheless, they hint at hidden factors affecting the labeling process in different institutions: while there are initially differences in labeling costs between the same worker groups of MD and SU (H_1), no such differences exist in the beginning between worker groups of the same institution (H_3).

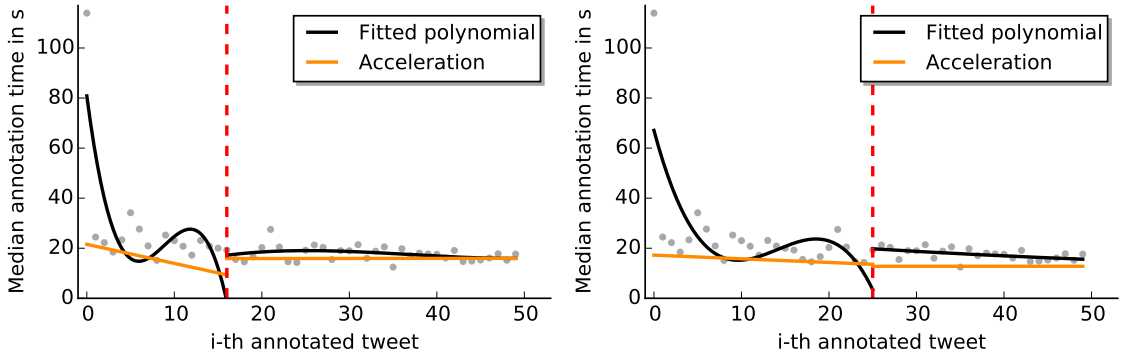


Figure 4.8: Fitted polynomials of degree three and their accelerations for M (S). Left: the interval boundary (red dashed line) is at $i = 16$ and the acceleration in the first interval is negative, so learning is still ongoing. Right: the interval boundary (red dashed line) is at $i = 25$ and the change in acceleration in the first interval is practically zero, so learning is completed.

4.3.3 Development of the Variance of Labeling Costs over Time

	MD			SU		
	Learn	*Rest*	*Fatigue*	*Learn*	*Rest*	*Fatigue*
Within	0.12	0.11 (0.11)	0.02	0.25	0.26 (0.29)	0.29
Between	0.65	0.04 (0.02)	0.04	1.84	0.07 (0.11)	0.15

Table 4.1: Between-subjects and within-subjects variability for the different institutions. Values in brackets are obtained when *Rest* of group M is split into *Rest* and *Fatigue*, otherwise only *Learn* and *Rest* are used.

After log-normal transforming the labeling costs in MD and SU, the resulting plots yield a Gaussian shape. We omit the plot to facilitate readability. Therefore, as described in Figure 4.2, we obtain two sublevels for each level. For group S, we set $i = 20$ according to Section 4.3.1, meaning tweets 1-20 are used in level *Learn* and tweets 21-50 in *Exploit*. For group M, $i = 40$ according to Section 4.3.1, so *Learn* comprises tweets 1-40 and *Exploit* utilizes tweets 41-150. We set $m = 80$ based on the personal feedback

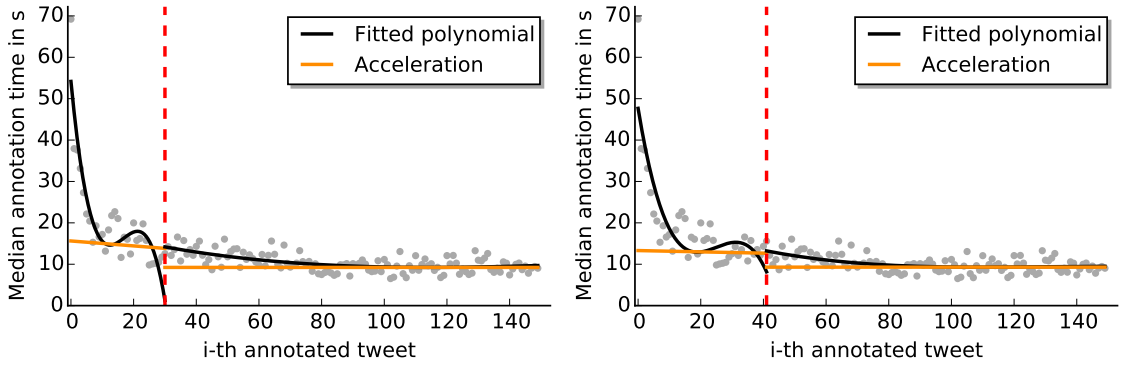


Figure 4.9: Fitted polynomials of degree three and their accelerations for SU (M). Left: the interval boundary (red dashed line) is at $i = 30$ and the change in acceleration in the first interval is negative, so learning is still ongoing. Right: the interval boundary (red dashed line) is at $i = 41$ and the change in acceleration in the first interval is practically zero, so learning is completed.

we received from few students of group M in SU after the annotation experiment because they mentioned that they felt tired and that they started to worry when the first students (of group S) started to leave. In Table 4.1, the resulting between-subjects and within-subjects variabilities of MD and SU are listed for the different levels.

In RQ1.3 we analyzed if the variance in the labeling costs decreased toward the end of workers' sessions. Our results indicate that the workers' labeling costs become more homogeneous after their learning phase since the between-subjects variability decreases from *Learn* to *Rest* and *Fatigue*. In addition, we find no indicators that workers in group M of MD fatigued, while there is potentially some weak evidence that workers in group M of SU might have fatigued as the between-subjects variability increases from *Rest* to *Fatigue*. This could also explain partially why in RQ1.2 workers of group M in SU labeled faster than their counterparts in MD. However, this interpretation needs to be examined more closely in the future.

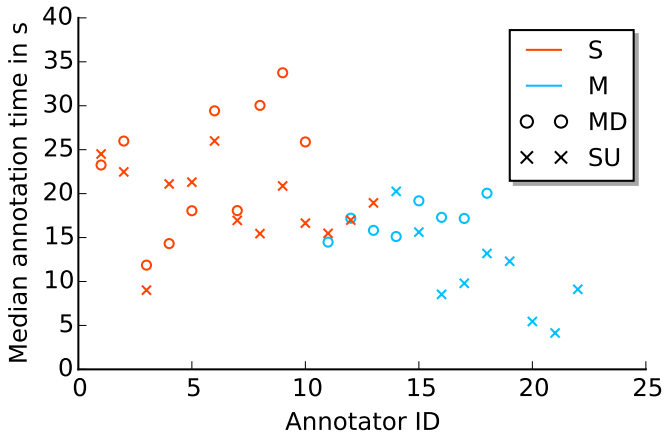


Figure 4.10: Median labeling costs per worker, sorted by worker groups and institutions.

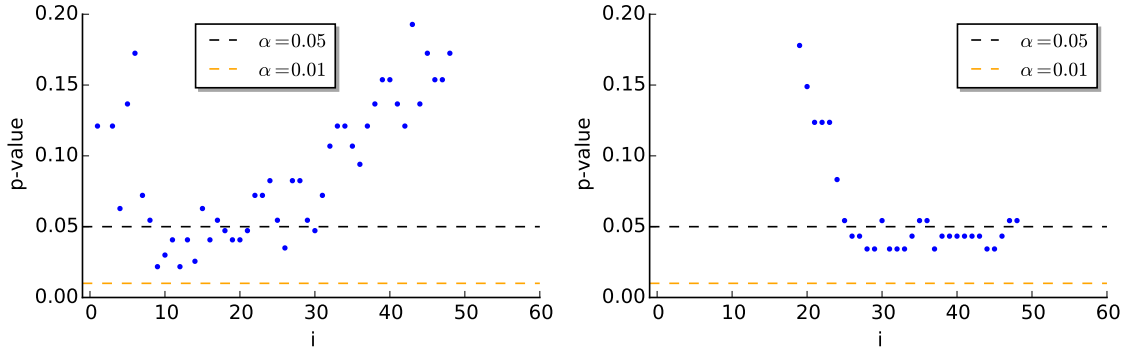


Figure 4.11: H_1 with i indicating the i^{th} tweet workers labeled. Left: MD (S) vs. SU (S). Right: MD (M) vs. SU (M). Whenever p-values for $k < 50$ are not displayed, they are larger than 0.2.

4.3.4 Development of Label Reliability over Time

Before reporting the results of examining the label reliability in the learning phase, we want to give an intuitive idea about what type of neighboring tweets kNN identifies for our given problem. We show in Figure 4.15 the most similar tweet for two tweets from our dataset. In both cases the unknown tweets share a partial quote with their nearest neighbors.

In our hierarchical classification task, we set i for the learning phase according to Section 4.3.1, i.e. for worker group S, i is set to 20 and n to 50, and for M, i is set to 40 and

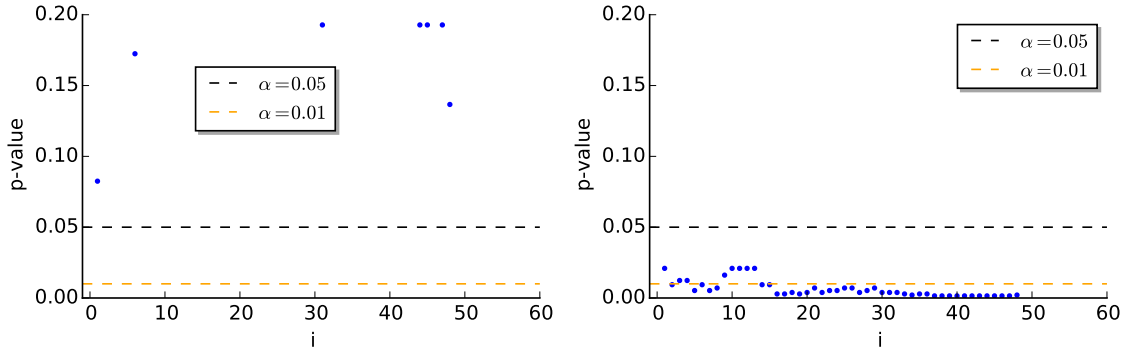


Figure 4.12: H_2 with i indicating the i^{th} tweet workers labeled. Left: MD (S) vs. SU (S). Right: MD (M) vs. SU (M). Whenever p-values for $k < 50$ are not displayed, they are larger than 0.2.

n to 150. The resulting training sets with these parameters are visualized in Figure 4.3. We show in Figure 4.16, the resulting hierarchical F1-scores for MD and SU. Regardless how we vary k, the number of neighbors used in kNN to predict unknown tweets in SU and MD, the predictors trained on tweets from a worker's exploitation phase consistently outperform the predictors trained on tweets from her learning phase. The performance gap between both predictors is larger when considering less neighbors in kNN for predictions, while it decreases when the number of considered neighbors increases. When increasing k beyond $k = 7$, the F1-scores remain the same as there are no further additional neighboring tweets available in the training set. Since the difference in F1-scores between $k = 1$ and $k = 9$ is marginal and forms almost a horizontal line, we argue that this illustrates that workers indeed learned a concept because otherwise there should have been some ups and downs in the scores.

In RQ1.4 we analyzed the reliability of labels obtained during learning and exploitation phase. Our result suggests that workers' labeling quality increases after the learning phase. The results of Section 4.3.2 also contribute to analyzing label reliability as explained in Section 4.4.

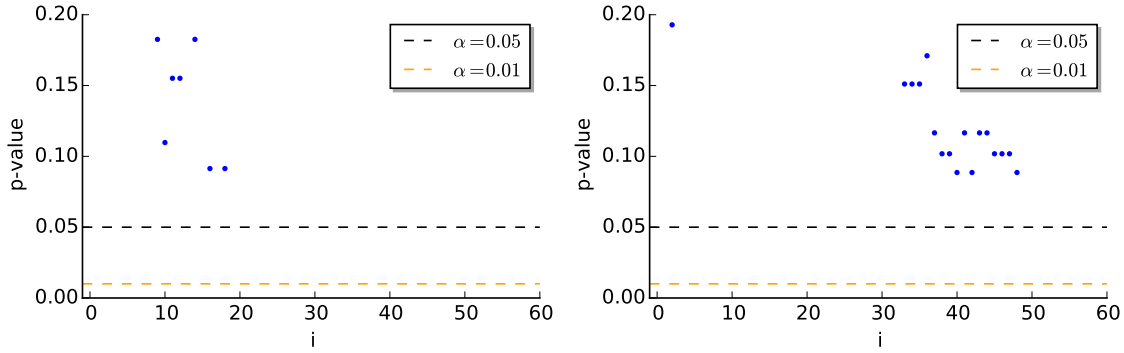


Figure 4.13: H_3 with i indicating the i^{th} tweet workers labeled. Left: MD (S) vs. MD (M). Right: SU (S) vs. SU (M). Whenever p-values for $k < 50$ are not displayed, they are larger than 0.2.

4.4 Discussion

4.4.1 Summary of Findings

We started this chapter with the expectation that workers would learn a conceptual model for each set of labels, so that labeling costs, in the form of total annotation time per tweet, would be higher during a crowd worker's learning phase and lower once the conceptual model was learned (exploitation phase). Our results give indications in support of this expectation: the labeling costs stabilize after some time on a lower level than observed at the beginning of each labeling session. Our results also indicate that this change from higher to lower labeling costs can be traced to some extent in both locations for the experiments, Sabancı (SU) and Magdeburg (MD). Furthermore, the reliability of labels assigned during a crowd worker's exploitation phase is higher than in her learning phase. All of these findings agree with the literature [7, 29, 30, 13] and common sense: when humans repeat a task multiple times they get better at it. Thus, our findings underscore the importance of training workers properly before they start an actual labeling task. In contrast to prior work, we determined in Sections 4.3.1 and 4.3.2 the length of a crowd worker's learning phase and quantified it in terms of how many documents she has labeled before. While doing so, we also found that the duration of the learning phase depends on the total num-

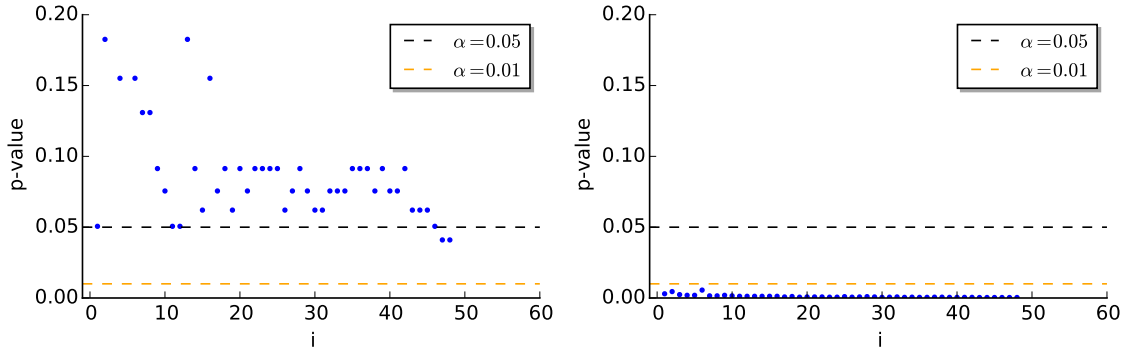


Figure 4.14: H_4 with i indicating the i^{th} tweet workers labeled. Left: MD (S) vs. MD (M). Right: SU (S) vs. SU (M). Whenever p-values for $k < 50$ are not displayed, they are larger than 0.2.

ber of tweets that the workers have been asked to label. Our observation indicates that the workers of tweet sets of size M had to see more tweets than those labeling a set of size S before they completed their learning phase. We suspect that this duration may have been influenced by the presence of a progress bar as part of the user interface of our annotation tool. On this bar, each worker could see how much each labeled tweet contributed to her annotation session. The progress bar operated as an incentive for workers labeling sets of size S: each labeled tweet incremented the progress bar substantially by two percent, which acted like an incentive, thus leading to closer attention and to the tendency to learn fast. Analyzing the metadata, e.g. gender, that our annotation tool collected about workers showed that they were roughly equally distributed across S and M, so they do not explain the observed differences. Therefore, we need to further investigate this effect.

According to Section 4.3.3, the labeling costs per tweet were more homogeneous in MD than in SU. In SU, the group labeling a set of M tweets had significantly lower labeling costs per tweet during the exploitation phase than those labeling S tweets. Likewise, the SU group labeling M tweets was significantly faster than their counterparts in MD. One possible explanation may be in the experiment design at the two locations. In SU, the workers of sets of size M were in the same room as the workers of sets of size S, and working simultaneously with them. Hence, when the latter, having to annotate less

Figure 4.15: Two examples of kNN using edit distance. The label of the upper tweet is to be predicted and the lower tweet represents its nearest neighbor.

tweets, were done, the former may have gotten an incentive to work faster. In MD, most workers of sets of size M had never worked simultaneously with those annotating sets of size S [2]. Therefore, it would be interesting to repeat the experiment, but having groups S and M use separate rooms. If this result of artificially speeding up the annotation process due to peer effects is reproducible, it would be interesting to analyze to what extent the labeling quality is affected. If it remains largely unaffected, this would offer a chance to reduce labeling costs by deliberately creating an environment similar to ours.

Our statistical tests indicated that initial labeling costs per tweet were identical between groups S and M at the same location. However, comparing the same groups across the locations showed significant differences in the initial labeling costs. One interpretation of this result is that there are certain hidden factors that we did not capture in our experimental setup. For example, personality traits like curiosity or motivation could affect the duration of a worker's learning phase.

In conclusion with regards to RQ1, we found that crowd workers undergo a learning phase which is followed by an exploitation phase. The learning phase is characterized by a quick drop in annotation times (labeling costs) and lower label reliability, while the annotation times in the exploitation phase converge to a stable level, that is lower than in

[2]The workers in MD used the same room, but at different times.

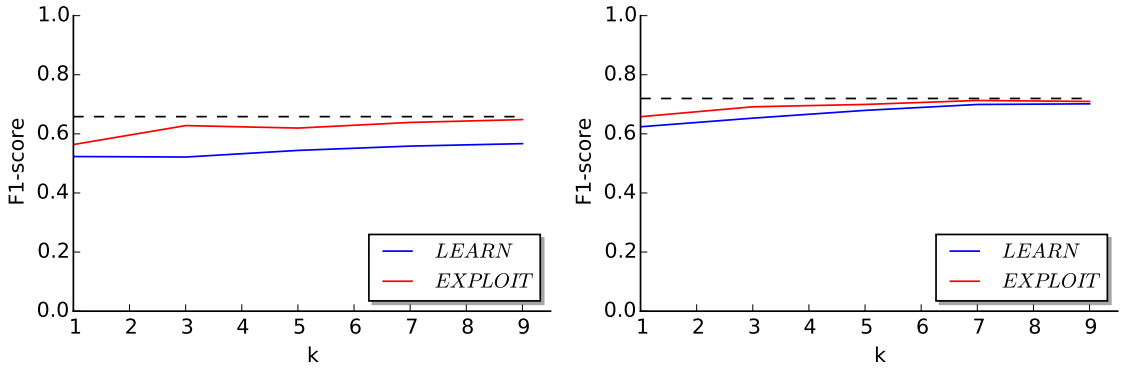


Figure 4.16: Hierarchical F1-scores for kNN predictors trained on tweets from the learning phase (”LEARNING”) and on tweets from the exploitation phase (”EXPLOIT”) when varying k. Left: MD. Right: SU.

the learning phase, and the label reliability is higher.

4.4.2 Applications

Knowing how many documents a crowd worker has labeled previously could be used as a proxy to estimate label reliability. For example, it should be incorporated into cost models in active machine learning, whose task is to estimate the costs for labeling unlabeled documents. This could improve the cost model's predictive capabilities. A promising candidate that could benefit from this idea is [30]. Similarly, it would be useful if crowdsourcing platforms stored, in addition to the assigned labels and annotation times, how many documents a worker has labeled previously for the current task because this would allow to estimate the reliability of her assigned labels. This information could then be utilized when aggregating crowdsourced labels to a ground truth by weighting each label according to its reliability. This could lead to more accurate labels than simple majority voting.

Since label reliability increases during a crowd worker's exploitation phase, it seems useful to discard labels that were assigned during the learning phase, or alternatively these documents should be relabeled during the exploitation phase as documents with less reliable labels affect predictors trained on such data negatively.

Knowing the duration of workers' learning phases is beneficial for designing training sessions of adequate length for specific labeling tasks. For example, the authors of [29] find that training crowd workers in general improves label reliability and efficiency, while in [36] it is found that person-oriented training strategies work particularly well. Combining these findings with the additional information about the duration of the learning phase, which is obtained by applying the methods from this chapter, it is possible to tailor micro-tasks for training to the learning phase of workers instead of fixing the length of such tasks arbitrarily. The methods we used for quantifying the length of the learning phase are applicable to any labeling task and our source code is publicly available[3]. To determine the length of the learning phase, one should first perform a preliminary study with few participants to acquire annotation times for the labeling task at hand and afterwards the length of the learning phase can be computed.

Furthermore, based on our findings, requesters in crowdsourcing should aim at keeping a high retention rate of workers in their micro-tasks because this ensures that workers reach their exploitation phase leading to mutual benefits - requesters receive high-quality labels, while workers can complete more tasks in a shorter time, leading to higher rewards. Employing a dynamic payment strategy, which yields higher rewards once workers reach their exploitation phase, could facilitate this scenario to keep workers motivated.

4.4.3 Generalizability and the Role of the Experimental Setup

Besides the possible impact of the progress bar in our annotation tool, potentially fatigued workers, and the slightly different experimental setup at the two locations, where workers either labeled at the same time or at different times, the subject of the tweets could also affect generalizability, because the US presidential election was a hot topic. In the next step, we first plan to replicate our findings using a different dataset, which also drew a lot of attention, namely the Turkish referendum in 2017. In the second step we would analyze a less hotly debated topic. Since we focused on a single labeling task, we also plan to use a crowdsourcing platform like Amazon Mechanical Turk to investigate if our

[3] `https://github.com/fensta/InfSci2017`

findings also hold for larger datasets and different types of labeling tasks.

4.4.4 Future Work

An open question about the workers' labeling behavior over time is how long gaps between annotation sessions could be such that workers still remember their learned conceptual models. To address this question, we used our Twitter dataset TRAIN comprising 500 tweets described in Section 3.1. Workers performed their labeling tasks in three separate sessions of maximum 90 minutes each. 150, 200, and 150 tweets were labeled in the first, second, and third session respectively. Workers had to take a break between each session for at least 30 minutes. Using three sessions allowed us to investigate if workers remembered their learned conceptual models from previous sessions or whether they had to relearn them. In MD one volunteer labeled those 500 tweets whereas in SU three workers completed this task. Our preliminary results suggest that workers still remember their conceptual models after having a break of at most a day in between sessions. However, after a break of three to four days in between sessions, we could observe that they had to relearn their models. This is because when they started new sessions, their labeling costs were initially high, but quickly converged back to the level of their previous sessions. Nevertheless, more data is needed to analyze this particular aspect in more depth.

Two other open questions are related to the conceptual model of workers. First, since the resulting F1-scores from Section 4.3.4 for $k = 3$ is comparable with the one for $k = 9$, it suggests that workers do not need to remember many tweets to learn their conceptual models. The question to be analyzed would then be how workers learn their conceptual models. A follow-up question would be how does such a conceptual model look like? In this chapter we assumed its existence and used kNN to model the expected worker behavior, i.e. that the workers' models are based on the similarity between the texts they read, so that they transfer the label from one text to another. Our results indicate that kNN may have been part of the workers' models. However, kNN is based on a formal notion of similarity, while people are known to be subjective about similarity [50, 51]. We plan to consider different ways of capturing tweet similarity to investigate how

these measures affect the temporal dynamics (i.e. the speedup in tweet labeling time) of workers' behavior.

While setting up our annotation experiment, we already had additional research questions in mind that could be analyzed with our annotated dataset[4]. For example, assessing the effect of ambiguous and non-ambiguous tweets on the annotation time and resulting predictor performance is an appealing topic for future work. The tweet ambiguity could be derived from worker disagreement in the same way as proposed in [11], where worker disagreement implies ambiguous tweets. If there are non-ambiguous and ambiguous tweets in a dataset – are the labels assigned to such tweets equally reliable during learning and exploitation phase? How do these tweets affect predictor performance? Is it maybe even possible to predict the ambiguity of tweets in advance? This way it might be reasonable for workers to not label ambiguous tweets at all because they might confuse not only workers, but also the predictors trained on such data. This would ultimately lead to a new crowdsourcing methodology which considers learning phase and exploitation phase of workers as well as the ambiguity of documents like tweets. The basis for this approach is knowledge about the interplay between tweet difficulty and label reliability, which will be discussed in the next chapter.

[4] https://www.researchgate.net/publication/325180810_Infsci2017_dataset

Chapter 5

Influence of Difficult Tweets on Annotation Behavior

To investigate RQ2, i.e. the extent to which tweet difficulty influences label reliability we perform a preliminary simulation experiment using the dataset described in Section 3.1. The main motivation for conducting a preliminary experiment on the existing dataset is that we consider the recruited crowd workers faithful, while this is more challenging on real crowdsourcing platforms like Amazon Mechanical Turk, where noise could mask a potential connection between both factors. After stating our assumptions and refined research questions in Section 5.1, we describe the methods we use for performing our experiment in Section 5.2 and report the results in Section 5.3. This is concluded by discussing the implications of our findings in Section 5.4. Parts of this chapter appeared in [52].

5.1 Introduction

As shown in Chapter 4, labels assigned by crowd workers become more reliable in their exploitation phase. Similarly, the time needed to assign labels to documents drops rapidly in a worker's learning phase until it converges to a roughly constant level in the exploitation phase. Since annotation times are typically associated with labeling costs, shorter

annotation times are preferred. Thus, when experimenters want to recruit workers on a crowdsourcing platform who are likely to assign high-quality labels, suitable workers should (a) have completed similar tasks before and (b) have reached the state where labeling costs are approximately constant to keep the time needed for task completion short.

In practice, however, we suspect that this strategy could be affected by the inherent difficulty of the documents to be labeled since some documents are more difficult to label than others. Therefore, we expect that labels assigned to difficult documents will be less reliable. Using these difficult documents for training could affect the performance of the resulting predictors adversely. In contrast, if the reliability of the labels in the training set is high, resulting predictors could improve their performance. Investigating this idea allows us to address RQ2 of this thesis.

Thus, we assume that label reliability can be inferred from measuring the performance of predictors: given the performances of two predictors, we assume that the one achieving better performance was trained on documents with more reliable labels. We define ”document difficulty” informally as the set of factors that determine to what extend workers are hesitant in choosing among the available labels for a document. These factors may be features of the document, e.g. words in the document, but may also be in the eye of the beholder, e.g. affected by the workers' perception of and attitude towards the subject matter. Since we cannot fix the factors making a document difficult as solely inherent to the document, we rather rely on difficulty indicators, which are labeling cost, worker disagreement [11] and predictor certainty [34].

Since modeling the difficulty of tweets has been rarely the subject of investigation, we use the dataset from Section 3.1 to study the interplay between tweet difficulty and label reliability in crowd workers' learning phase and exploitation phase. To the best of our knowledge, this problem has not been studied before.

Therefore, we investigate the following research questions:

- **RQ2.1**. How does document difficulty in the training set affect the performance of resulting predictors in the learning phase and in the exploitation phase?
- **RQ2.2**. Are these effects from RQ2.1 meaningful?

Our analysis should be considered as a preliminary experiment to see if any interesting connection between tweet difficulty and label reliability exists. If there is a connection, in the next step real crowdsourcing experiments can be performed. This is a common approach in crowdsourcing, e.g. [53, 54, 55], for multiple reasons. For one, budget may be saved if proposed methods turn out not to work. Another reason is that one might want to run an experiment first in a controlled environment to avoid external influence factors which cannot be ensured in crowdsourcing.

5.2 Methods for Analysis

This section describes the methods we employ for designing our experiment to address the two research questions.

5.2.1 Modeling Crowd Workers and Tweets

In our analysis we utilize TRAIN from Section 3.1, i.e. we use the datasets from both geographical regions, Magdeburg (MD) and Sabancı (SU), with 500 tweets labeled hierarchically in terms of sentiment. Therefore, each tweet is either *Relevant* or *Irrelevant* and *Factual* or *Non-factual*. If a tweet is considered *Non-factual*, it is also either *Positive* or *Negative*. Similar to Section 3.1.5, additional labels and annotation times of *Irrelevant* tweets are ignored.

5.2.2 Modeling Tweet Difficulty

In the remainder of this chapter, we refer to difficult tweets as ambiguous tweets and to the remaining, easier tweets as non-ambiguous tweets. Since there is no ground truth for tweet difficulty available, we approximate the difficulty of a tweet t by computing its difficulty score DS. $DS(t)$ combines three heuristics, namely worker agreement (A) [11], predictor certainty (PC) [39], and labeling cost (L):

$$DS(t) = A + PC + L \tag{5.1}$$

where $A, PC, L \in [0, 1]$. We define higher difficulty scores in this equation to correspond to non-ambiguous tweets.

Using worker agreement in Equation 5.1 instead of worker disagreement is more appropriate because, according to our definition, higher difficulty scores imply non-ambiguous tweets. For determining the worker agreement A of tweet t, we devise a scoring function $A(t)$ yielding values between 0 (no agreement) and 1 (perfect agreement). The worker agreement of each hierarchy level must contribute to A. Specifically, we use majority voting to assign a label to each hierarchy level. A level should contribute more to A if more workers agreed on the label. Since lower hierarchy levels might have been labeled by less workers than the first level (namely if workers deemed a tweet *Irrelevant* or *Factual*), higher levels tend to contribute more to A. This reasoning is reflected in the following equation:

$$A(t) = \sum_{i \in \ Levels} \frac{|workers_{maj}|}{|workers_i|} * \frac{|workers_{maj}|}{total_{maj}} \tag{5.2}$$

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 5.2 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. The following two examples illustrate how Equation 5.2 approximates worker agreement. First, 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*

Therefore, the majority labels for $t1$ are *Relevant*, *Non-factual*, and *Negative*, leading to $A(t1) = 4/4 * 4/9 + 3/4 * 3/9 + 2/2 * 2/9 = 0.92$. In total, nine workers assigned the majority labels (four on the first level, three on the second level, two on the third level), so $total_{maj} = 9$. In the second example, suppose there was a tie on the second level of $t1$, i.e.

- First hierarchy level: *Relevant*, *Relevant*, *Relevant*, *Relevant*
- Second hierarchy level: *Factual*, *Non-factual*, *Non-factual*, *Factual*
- Third hierarchy level: -, *Negative*, *Negative*, -

This time there are two possibilities for the majority labels: either *Relevant* and *Factual* or *Relevant*, *Non-factual*, and *Negative*. In this case the majority labels would be chosen randomly. Suppose the latter one is chosen; then the resulting worker disagreement score would be $A(t1) = 4/4 * 4/9 + 2/4 * 2/9 + 2/2 * 2/9 = 0.78$. Note that in this case $total_{maj} = 9$ instead of $total_{maj} = 8$ because exactly one tie occurred on the second hierarchy level, leading to a lower agreement score than in the first example. Converting worker agreement $A(t)$ for tweet t into worker disagreement $DA(t)$ is accomplished in a straightforward manner:

$$DA(t) = 1 - A(t) \tag{5.3}$$

The resulting values of DA will be again between 0 (perfect agreement) and 1 (no agreement), but compared to A, the meaning of the values has now switched.

A higher predictor certainty PC for a tweet indicates non-ambiguous tweets. To compute it, we build a kNN[1] predictor for each worker separately since sentiment is subjective. The predictor is trained on 40% of a worker's labeled tweets and the longest common substring[2] according to Equation 3.1 is used to compute the similarity between any pair of

[1] We opted for kNN as it considers neighborhoods and we believe that the type of difficulty we investigate is a local phenomenon ("Are similar tweets difficult or easy to label?"), so we do not want to use an SVM or similar predictors as they learn globally optimal models ("Is the tweet easy or difficult to label?"); the latter could be investigated in the future separately.

[2] We obtained similar results when choosing edit distance or longest common subsequence.

tweets. Since kNN does not naturally provide a certainty for the predicted label j of tweet t, we approximate it as follows:

$$certainty_j(t) = \frac{n_j + s}{k + c} \tag{5.4}$$

where n_j is the number of the k neighbors that share label j, s being a smoothing factor to avoid zero probabilities, and c being the number of possible classes that exist on a certain hierarchy level. In our experiment we set $s = 1$. We store for each tweet of a worker's test set (60% of the labeled tweets) the certainty PC of the predicted labels. Repeating this process for all workers yields a list of predictions per tweet on each hierarchy level. To obtain a single certainty per tweet, we first average the certainties (of the different workers who labeled the tweet) per level and from these certainties we pick the maximum certainty per level, i.e. this process yields three values. Each of these three certainties corresponds to the predicted majority label on the respective hierarchy level. Averaging these three values yields $PC(t)$. This procedure is reflected in the following equation:

$$PC(t) = \frac{1}{3} \sum_{i \in Levels} \max_{j \in Labeled} \frac{\sum_{k \in Workers} certainty_j(t)}{|Workers|} \tag{5.5}$$

where $Labeled$ is the set of predicted labels for t on hierarchy level i, $Workers$ is the set of crowd workers who labeled t in their test sets, and $Levels$ is the set of hierarchy levels in the labeling scheme, in our case $Levels = \{1, 2, 3\}$. Note that in this procedure we are not accessing the sentiment labels which kNN predicts for a tweet. Instead, we only use the predictor certainties of the sentiment labels that kNN assigned to the tweets. Therefore, we are not leaking any information about the actual sentiment labels to the sentiment predictors (cf. Section 5.2.5). Table 5.1 illustrates how $PC(t1)$ is obtained for $t1$. In this case two workers have $t1$ in their test set, hence we have four predictor certainties (two predicted labels per worker) per hierarchy level. For example, kNN is 80% certain, according to Equation 5.4, that worker 1 (first row, first column) would assign *Relevant* to $t1$ on the first hierarchy level. In contrast, kNN is only 20% certain for her to assign *Irrelevant*. The certainties are averaged per label and per level (row 3), e.g. the average certainty of kNN to assign *Relevant* on the first hierarchy level is

$(80\% + 70\%)/2 = 75\%$, while it is $(20\% + 30\%)/2 = 25\%$ for *Irrelevant*. Averaging these three remaining certainties results in $PC(t1) = 68\%$.

	First level	Second level	Third level
Worker 1	(R, .8), (IR, .2)	(F, .4), (NF, .6)	(P, .3), (N, .7)
Worker 2	(R, .7), (IR, .3)	(F, .2), (NF, .8)	(P, .5), (N, .5)
Avg. certainty	(R, .75), (IR, .25)	(P, .5), (N, .5)	(P, .4), (N, .6)
Maximum certainty	.75	.7	.6
$PC(t1)$		$(.75 + .7 + .6)/3 = .68$	

Table 5.1: Example how Equation 5.5 aggregates the predicted certainties for tweet $t1$. The columns represent the hierarchy levels in the labeling task. We use the following acronyms to represent the predicted sentiment labels: R: *Relevant*, IR: *Irrelevant*, F: *Factual*, NF: *Non-factual*, P: *Positive*, N: *Negative*. Suppose two workers labeled $t1$ in their test sets and kNN predicted for each worker a tuple of (sentiment label, certainty) according to Equation 5.4 per hierarchy level. "Avg. certainty" averages the predicted certainties per label per hierarchy level. "Maximum certainty" shows which certainty would be kept according to Equation 5.5 and the last row shows the final result of the computation, thus $PC(t1) = 0.68$ in this case.

The labeling cost L for tweet t corresponds to t's median annotation time. The higher it is, the more ambiguous a tweet is. However, since high values of $DS(t)$ are associated with non-ambiguous tweets, L must be inverted. We choose as labeling cost for t the median annotation time across all workers who labeled it. The median is more appropriate than the average in our case due to its robustness toward outliers because some workers had a few random spikes in their annotation times. After normalizing the labeling cost, the following equation follows:

$$L(t) = 1 - \frac{cost_t - cost_{min}}{cost_{max} - cost_{min}} \tag{5.6}$$

where $cost_t$ is the median labeling cost of tweet t, $cost_{min}$ ($cost_{max}$) is the lowest (highest) median labeling cost across all tweets.

After computing DS for each tweet, we apply k-means with $k = 2$ to cluster the difficulty scores. Each tweet is now assigned a difficulty label according to its cluster membership – either *Disagreement*, which indicates difficult tweets, or *No Disagreement* for the remaining tweets.

5.2.3 Design of the Simulation Experiment

By training predictors we want to answer RQ2.1, i.e. if difficult tweets affect label reliability in the learning phase and in the exploitation phase. We use the dataset described in Section 3.1 to simulate the effect of tweet difficulty on label reliability. The goal is to predict the hierarchical sentiment labels (*Relevant*, *Irrelevant*, *Factual*, *Non-factual*, *Positive*, *Negative*) according to Section 3.1.2. We measure predictor performance in terms of hierarchical F1-score, which is recommended by Kiritchenko et al. for hierarchical labeling tasks [48]. Specifically, we analyze the effect of the following independent variables on predictor performance:

- *difficulty*: ambiguous (difficult) or non-ambiguous (easy) tweets
- *phase*: learning phase or exploitation phase
- *training set size*: number of tweets in the training set
- *neighbors*: number of nearest neighbors in kNN
- *institution*: either MD or SU

We expect meaningful patterns observed in this simulation to hold despite varying the abovementioned variables. Otherwise the patterns might be due to chance. For example, if one predictor outperforms another one, this result should hold even if the size of the training set changes.

The core assumption in this simulation experiment is that the reliability of labels can be inferred from measuring the performance of trained predictors: if predictors achieve higher F1-scores, the sentiment labels in their training sets are considered more reliable.

In other words, we use F1-score as a proxy for the reliability of labels. Therefore, we train two predictors per crowd worker, PredND trained only on easy, i.e. non-ambiguous, tweets and PredD which is trained solely on difficult, i.e. ambiguous, tweets. We fix all of the abovementioned variables, so that only the variable *difficulty* of the training set differs between both predictors. This allows us to draw conclusions about the effect of tweet difficulty on label reliability.

5.2.4 Learning Phase & Exploitation Phase in Worker Behavior

For the experiment in Section 5.2.5, our dependent variable, predictor performance, is affected by two parameters: the number of tweets used in the training set and tweet difficulty. That means we plot a curve of the predictor performances once for ambiguous and once for non-ambiguous tweets while varying the number of tweets in the training set. However, workers undergo a learning phase [56, 7, 46], i.e. a drop in annotation times occurs in the beginning of an annotation session. Thus, the phase – either learning phase or exploitation phase– is also an independent variable that we need to control for in our experiment. Therefore we perform the experiment once for the learning phase and once for the exploitation phase because within these phases the annotation times can be considered similar.

Originally, workers labeled either S, M, or L tweets of TRAIN according to their worker group in Section 3.1.2 and we found in Section 4.3.1 that the length of the learning phase differs across the worker groups. To avoid having to control for this variable as well, i.e. repeating the experiment with the two phases once for each worker group, we fix the length of the learning phase across all three worker groups. When aggregating all annotation times per institution, either MD or SU, we obtain for the length of the learning phase approximately 25 tweets, i.e. the first 25 labeled tweets of each worker are used for their learning phase and their next 25 labeled tweets are utilized for their exploitation phase to have a balanced experimental setup. Therefore, we use in total the first 50 labeled tweets of each crowd worker in both institutions. Any other labeled tweets are discarded. Another reason for not using more tweets for the exploitation phase is

to avoid uncontrollable side effects such as fatigue because in Section 4.3.3 we found possible indicators for fatigued workers.

5.2.5 Building Predictors

One sentiment predictor (kNN) is trained per crowd worker in MD and SU because sentiment analysis is subjective. The exact training procedure of PredND and PredD for a single crowd worker is illustrated schematically in Figure 5.1. The training set (containing only ambiguous or only non-ambiguoustweets) is derived from tweets 1-25 in the learning phase and once from tweets 26-50 in the exploitation phase. This leads effectively to four datasets per worker to which we refer in the remainder as strata, namely:

1. LEARN_NONAMBIGUOUS: non-ambiguous tweets that were labeled in a worker's learning phase

2. LEARN_AMBIGUOUS: ambiguous tweets that were labeled in a worker's learning phase

3. EXPLOIT_NONAMBIGUOUS: non-ambiguous tweets that were labeled in a worker's exploitation phase

4. EXPLOIT_AMBIGUOUS: ambiguous tweets that were labeled in a worker's exploitation phase

Hierarchical learning is performed by training in total six predictors (two predictors are trained per level and we have three levels). Note that we introduced an extra label besides the sentiment labels to indicate that no label exists on a certain hierarchy level. This is necessary as *Irrelevant* tweets have only a label on the top-most hierarchy level. To assess the performance of the trained predictors in terms of hierarchical F1-scores (micro-averaged over all workers in a stratum), the labels of the remaining tweets in a worker's stratum are estimated per hierarchy level. For example, if PredND is trained on five tweets that a worker labeled in LEARN_NONAMBIGUOUS, it will be evaluated on her remaining 20 labeled tweets.

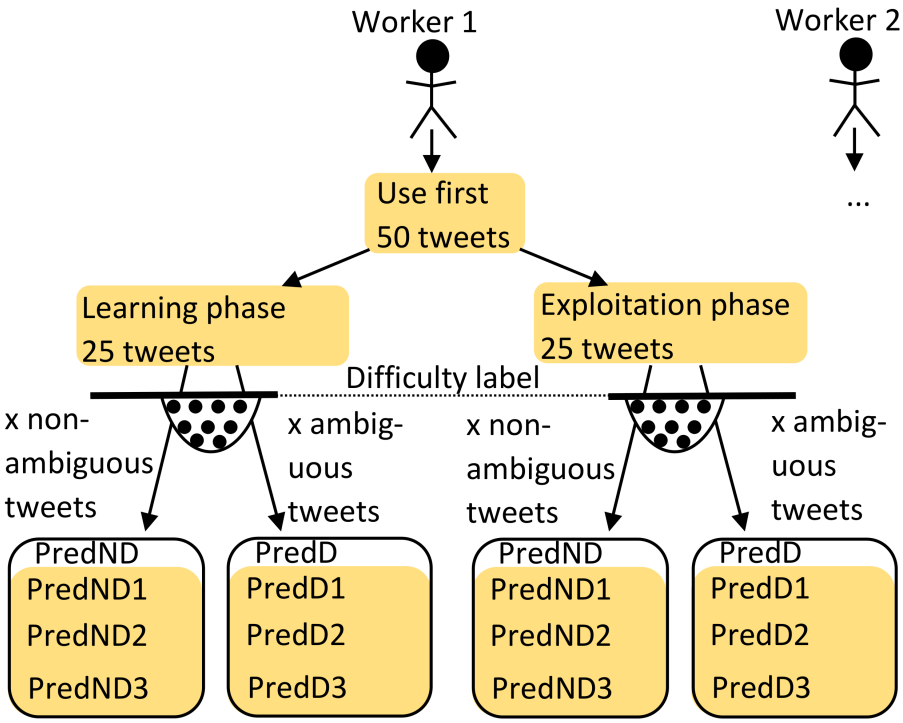


Figure 5.1: Overview how predictors, using x tweets for training, are built for a single crowd worker.

5.2.6 Testing the Meaningfulness of Observed Patterns

Since we vary many parameters in our simulation (see Section 5.2.3), it will be hard to depict all plotted configurations. Instead, our main goal is to identify patterns that hold over different configurations as these are more likely to be meaningful. We will report all our results in an encoded form to make finding patterns more straightforward. Instead of showing how the F1-scores of the predictors develop when varying the size of the training set, we simply state if one of the two resulting F1-curves dominates the other one. In that case there are three possible outcomes: either curve dominates the other one or there is a tie. The details about the encoding are explained in Section 5.3.2. However, reporting these encoded results permits us to test if there are significant differences in the proportions of the three outcomes using the two-tailed Fisher's exact test (cf. Appendix A.3). Fisher's exact test (instead of a chi-square test) is suitable since some of the outcomes occur rarely.

5.3 Results of Analysis

First we report the observed patterns of the simulation experiment and then we address their meaningfulness, that is how likely they occurred by chance.

5.3.1 Observed Patterns in the Simulation Experiment

This section addresses RQ2.1. In our dataset, non-ambiguous and ambiguous tweets are roughly equally distributed, with non-ambiguous tweets (according to Eq. 5.1) accounting for 50% to 57% of the tweets depending on the stratum as illustrated in Table 5.2. That means the classes are sufficiently balanced, thus there is no need to take any special countermeasures in the classification task.

	MD	SU
non-ambiguous	68 (50.4%)	93 (57.4%)
ambiguous	67 (49.6%)	69 (42.6%)

Learning phase

	MD	SU
non-ambiguous	78 (55.3%)	86 (54.3%)
ambiguous	63 (44.7%)	72 (45.7%)

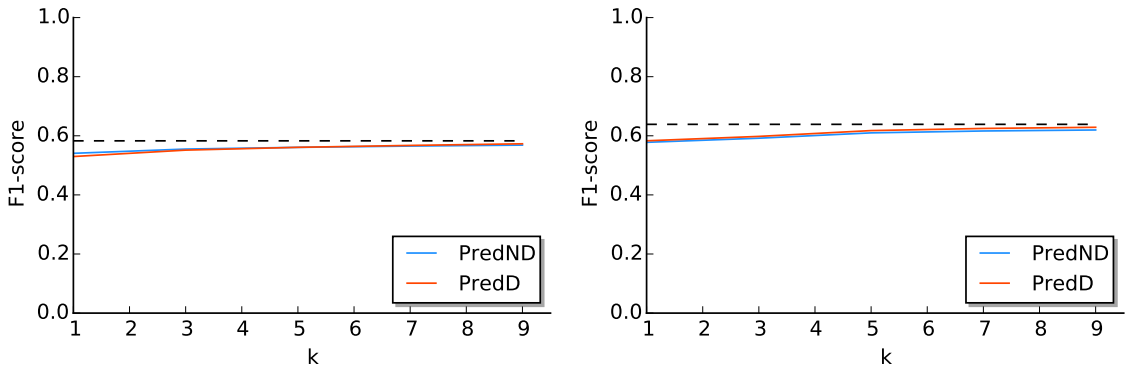
Exploitation phase

Table 5.2: Absolute numbers and percentages of non-ambiguous/ambiguous tweets per stratum for both groups, MD and SU.

First, we show some sample F1-curves of the trained predictors because afterwards we encode them into a compressed form to be able to report all of our results. This allows to identify certain trends whose statistical significance we examine thereafter.

We show the F1-curves of the kNN predictors trained on eight tweets per worker for the four strata while varying k, the number of neighbors in kNN. The predictors utilize

edit distance as a similarity metric. In Figure 5.2, the F1-curves of PredND trained on LEARN_NONAMBIGUOUS and PredD trained on LEARN_AMBIGUOUS are shown for MD and SU. In that case both predictors perform equally well. This observation holds in both groups and will be encoded as (T)ie in the compressed form. We note that the differences between the F1-curves in the learning phase are generally small. The corresponding F1-scores for the exploitation phase of MD and SU are depicted in Figure 5.3 using the same setup as described before. This means that now the performances of PredND trained on EXPLOIT_NONAMBIGUOUS and PredD trained on EXPLOIT_AMBIGUOUS are evaluated. This time, PredND outperforms PredD. This behavior is consistent in MD and SU and will be encoded as (N)o disagreement in the compressed representation. In this specific case, the F1-scores of PredND in SU are between 1.5% and 4.5% higher than in PredD. In MD, PredND achieves between 2% and 6% better F1-scores than PredD. We note that the differences between the F1-curves tend to be larger if PredND outperforms PredD. If PredD wins, both F1-curves are close to each other. In both figures it seems that considering more neighbors for predictions mainly improves the F1-scores of PredD but not PredND. This could indicate that less workers are necessary to label non-ambiguous tweets as opposed to ambiguous ones.

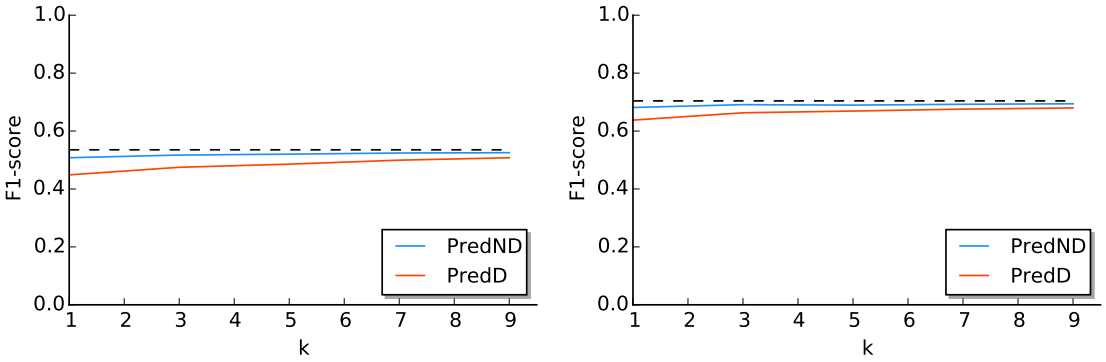


(a) MD

(b) SU

Figure 5.2: F1-scores of kNN with varying k. For each worker the training set comprises eight (non-ambiguous/ambiguous) tweets of the learning phase.

We report the outcomes of the remaining F1-curves of the predictors for the four strata



(a) MD (b) SU

Figure 5.3: F1-scores of kNN with varying k. For each worker the training set comprises eight (non-ambiguous/ambiguous) tweets of the exploitation phase.

with varying training sets containing between two and ten tweets as follows. At all times we compare in a stratum the F1-scores of PredND and PredD while varying k. We encode each outcome as follows (abbreviation in parentheses):

- (T)ie (both predictors exhibit the same F1-scores),
- (N)o disagreement (PredND outperforms PredD),
- (D)isagreement (PredD outperforms PredND).

We determine if predictor A outperforms predictor B by visually inspecting both F1-curves and if one curve yields a higher F1-score for some k and its scores are not lower than B's for all other k, A is considered to outperform B.

Each table contains the encoded outcomes over training sets comprising between two and ten tweets using different distance metrics. More specifically, Table 5.3 depicts the outcomes for the edit distance, Table 5.4 shows the outcomes for the longest common subsequence, and Table 5.5 gives the results for the longest common substring. One tendency in these tables is that the likelihood of seeing T drops as the number of tweets used for learning increases. We suspect that this phenomenon occurs because a small number of training tweets leads to a poor predictor performance anyway, no matter whether these

tweets were non-ambiguous or ambiguous. As soon as the number of training tweets increases, the difference becomes apparent, whereupon it becomes more likely that PredND is the best one.

We juxtaposed the winner predictors between the two groups MD and SU once for the learning phase and once for the exploitation phase. The numbers are too small to deliver robust results, but we observe a general tendency: PredND is more often the winner in the exploitation phase for SU than for MD. This could be seen as an indication that SU learned faster, but the phenomenon can also be explained by differences in size between the two groups: MD is smaller and thus more vulnerable to variations in the performance of the individual workers. Another related pattern across all groups is that T occurs frequently in the learning phase, while N tends to appear more often in the exploitation phase.

Phase \ #Tweets	2	3	4	5	6	7	8	9	10
Learning phase	T	T	T	D	D	D	N	N	N
Exploitation phase	T	T	T	N	N	N	N	N	N

MD

Phase \ #Tweets	2	3	4	5	6	7	8	9	10
Learning phase	T	T	T	T	T	D	D	T	N
Exploitation phase	T	N	N	N	N	N	N	N	N

SU

Table 5.3: Outcomes for the different strata using kNN with edit distance and a varying number of tweets in the training set of each worker.

Phase \ #Tweets	2	3	4	5	6	7	8	9	10
Learning phase	T	T	T	D	T	T	N	N	N
Exploitation phase	T	D	T	T	T	N	N	N	N

MD

Phase \ #Tweets	2	3	4	5	6	7	8	9	10
Learning phase	T	T	T	T	T	D	D	T	E
Exploitation phase	T	N	N	D	N	D	N	N	N

SU

Table 5.4: Outcomes for the different strata using kNN with longest common subsequence and a varying number of tweets in the training set of each worker.

5.3.2 Significance of Observed Patterns

To analyze the meaningfulness of these patterns according to RQ2.2, we run the two-tailed Fisher's exact test to see if the differences in the proportions of the outcomes are significant as described in Section 5.2.6. For comparing all pairwise proportions, our null hypotheses to be tested are: there is no difference in the proportion of N and D (T and N) (T and D) between learning phase and exploitation phase. The proportions are displayed in Table 5.6 and were obtained by adding up the outcomes from Tables 5.3-5.5. Using $\alpha = 0.05$ as significance level , we obtain the following results.

The proportions of N and T are significantly different in the learning phase and exploitation phase ($p < 0.0001$). This suggests that ties between predictors occur more frequently in the learning phase, while PredND outperforms PredD significantly more often in the exploitation phase. Likewise, the proportions of N and D differ significantly ($p < 0.02$) across both phases, which means that neither of PredND nor PredD wins sig-

Phase \ #Tweets	2	3	4	5	6	7	8	9	10
Learning phase	T	T	T	T	T	N	N	N	N
Exploitation phase	T	D	T	N	N	T	N	N	N

MD

Phase \ #Tweets	2	3	4	5	6	7	8	9	10
Learning phase	T	T	T	T	T	D	D	T	N
Exploitation phase	T	N	N	N	N	D	N	N	N

SU

Table 5.5: Outcomes for the different strata using kNN with longest common substring and a varying number of tweets in the training set of each worker.

nificantly more frequently in the learning phase, while in the exploitation phase PredND outperforms PredD significantly more often. When it comes to the proportions of T and D, no significant differences exist in the proportions ($p > 0.5$). Thus, the significance tests confirm our intuition about the existing patterns in the results, namely that T occurs mainly in the learning phase, N in the exploitation phase and D appears rarely in both phases.

5.4 Discussion

The results of our preliminary study in this chapter suggest for RQ2 that there is indeed a connection between the difficulty of tweets and the reliability of the labels that workers assigned to them. More specifically, the label reliability of easy, non-ambiguous tweets seems higher, because predictors trained on them achieve higher F1-scores. However, this holds only for a worker's exploitation phase, i.e. after workers have already labeled

	LEARN	EXPLOIT
T	31	12
N	13	36

N vs. T

	LEARN	EXPLOIT
N	13	36
D	10	6

N vs. D

	LEARN	EXPLOIT
T	31	12
D	10	6

T vs. D

Table 5.6: Occurrences of the encoded outcomes in a worker's learning (LEARN) and exploitation (EXPLOIT) phase.

some tweets (25 tweets in this work). In the learning phase, i.e. for the first 25 tweets, our results do not show any evidence for such a relationship. One possible explanation for this result could be that the labels workers assign in their learning phase [13, 7, 29, 56, 46] are generally of lower quality during that period [56, 46] as shown in Chapter 4. Therefore, the higher level of low-quality labels in the learning phase could be masking the effect of tweet difficulty on label reliability in a worker's learning phase.

It would be interesting to examine this hypothesis using a slightly different experiment setup than our current one in a new study: first, workers complete a labeling task in their first annotation session (same setup as in Section 3.1) and after a short break, they repeat the task with new tweets in a second session. If the noisy, low-quality labels due to the learning phase masked the relationship between tweet difficulty and label reliability in the learning phase of the first session, in the second session we would expect to see a pattern similar to the one we reported for the exploitation phase in this chapter, because workers should not have to go through another learning phase, assuming the break between two sessions is not too long. However, given that crowd workers tend to complete many micro-tasks, they will quickly reach their exploitation phase, meaning that labeling easier

tweets will increase the reliability of assigned labels in practice.

This motivates the idea of devising a tweet difficulty predictor to estimate the difficulty of unknown tweets for which a host of applications exist. We plan to apply this predictor as a filter before an actual crowdsourcing task. Given a large dataset, one could crowdsource a small seed first to train the difficulty predictor. It then estimates the level of difficulty in the unlabeled dataset and only tweets which are estimated to be easy would be crowdsourced. This idea is explored further in Chapter 6. Building such a difficulty predictor on a small seed set would also benefit active machine learning techniques, as they could be invoked only on easy tweets to obtain reliable labels from experts. Here the difficulty predictor would be used before invoking an active machine learning algorithm only for easy tweets. Furthermore, incorporating tweet difficulty into cost models in active machine learning, that estimate the costs for acquiring labels for unlabeled tweets, could enhance the models' accuracy.

Reducing the dataset size by filtering out difficult tweets could potentially increase the retention rate of the crowdsourcing task as workers might become less frustrated since micro-tasks can be completed with more ease. Furthermore, crowdsourcing a smaller dataset could save budget that will not be spent on difficult tweets. Even more budget could be saved if less crowd workers would be allocated to easy tweets, similar to [11]. Another way of using such a tweet difficulty predictor would be to assign easy tweets for labeling to inexperienced workers and difficult ones to experts [57]. The associated monetary compensation could possibly also vary depending on the level of expertise of crowd workers. This is related to the problem of optimal task routing in crowdsourcing where suitable workers should be identified for micro-tasks. For example, in [45] workers' cognitive abilities are used to match them to suitable tasks. This works for language fluency and visual tasks, but has not been tested for other types of tasks, such as sentiment analysis. If tweets are involved, a tweet difficulty predictor could complement this approach.

We note several limitations in our preliminary study. First, we used a relatively small dataset. Nevertheless, the tweets we used were diverse and we performed our experiment

independently in two different locations. Second, we investigated a single labeling task and it could bias the results. For example, in other tasks easy tweets might not be diverse enough to train good predictors. However, if sufficiently diverse tweets exist for a labeling task, we believe that our results will hold. Third, we evaluated only one predictor, kNN. Thus, replicating this experiment on a larger scale with more diverse predictors would help establish our findings.

To be able to utilize labels assigned by crowd workers during their training session, one should provide non-ambiguous tweets as the label reliability for this kind of documents is higher than for ambiguous ones. Once workers have reached their exploitation phase, they have a stable conceptual model which allows them to label more reliably and faster. Our dataset[3] and source code[4] are publicly available

Building on the finding that tweet difficulty affects label reliability, we introduce in the next chapter a multi-stage approach that separates difficult tweets from the rest using a difficulty predictor.

[3] https://www.researchgate.net/publication/325180810_Infsci2017_dataset

[4] https://github.com/fensta/PrelimStudy

Chapter 6

Predicting Tweet Difficulty

This chapter addresses RQ3 by proposing a new crowdsourcing methodology that takes the difficulty of documents (here: tweets) into account to produce more reliable crowdsourced datasets. This methodology utilizes the findings from Chapter 4 and Chapter 5. In Section 6.1 we explain our assumptions and the refined research questions to address RQ3. While Section 6.2 explains our crowd sourcing methodology, Section 6.3 describes the datasets we acquired for evaluation. Then Section 6.4 tests the feasibility of the proposed methodology and Section 6.5 points out open questions and possible future improvements. Parts of this chapter appeared in [58].

6.1 Introduction

Crowdsourcing is a popular mechanism to obtain large-scale labeled datasets 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 [46] or consistency [18], have been proposed for workers. Consistency might be affected by training, expertise, or fatigue emerging during a crowdsourcing task. In [19], 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 [20] (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 and the labels they assign. 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). Task-related properties such as a clear task specification [10] also improve label reliability. However, in our preliminary study in Chapter 5, we showed that the success of a crowdsourcing task also depends on properties of the documents to be labeled by the workers, specifically that document difficulty affects label reliability. 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 indeed be 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 are difficult to label which would waste worker budget) if labeled.

Our contribution is a new crowdsourcing methodology that a) improves the reliability of crowdsourced datasets and b) enhances the predictor performance that is learned on those datasets. Our method is based on the assumption that tweet difficulty can be derived from worker disagreement, i.e. the more workers disagree on the label of a tweet, the more difficult we consider it. This reasoning is in line with Aroyo et al. who argued that ”[crowd worker] disagreement is not noise, but signal” [33]. Based on this reasoning, 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 dataset and all documents with worker disagreement are considered ambiguous (difficult) 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. Evaluating this approach lets us address RQ3, i.e. how document (here: tweet) difficulty can be leveraged to improve the label reliability in crowdsourced datasets. In terms of In light of the above discussion, we refine RQ3 by studying the following research questions in this chapter:

- **RQ3.1**. How does the disagreement predictor perform?
- **RQ3.2**. Does the disagreement predictor improve gradually?
- **RQ3.3**. What is the effect of ambiguous tweets on sentiment classification?
- **RQ3.4**. What is the effect of allocating more budget to ambiguous tweets on sentiment classification?

While the first two RQs deal directly with our devised predictor, the last two RQs examine the overall potential of our approach given that it is feasible to predict worker disagreement. Hence, we address RQ3.3 and RQ3.4 by conducting simulations.

Unlike existing studies that have investigated the link between document difficulty and label reliability in crowdsourcing [44], 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 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 the authors of [59] 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.

6.2 Methods for Analysis

We propose a multi-stage iterative methodology, which is depicted in Figure 6.1. Given an unlabeled dataset U, we start with a small, randomly sampled seed set (see top part of Figure 6.1) to be labeled by the crowd workers w.r.t. a certain labeling task, e.g. sentiment analysis (see top-right corner of Figure 6.1). 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 Figure 6.1). Then, we train a disagreement predictor (see bottom-right corner of Figure 6.1) which estimates the worker disagreement in the unlabeled documents. Documents on which workers are expected to agree are moved to dataset C. Otherwise they are moved to dataset 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 Figure 6.1), thus refining the disagreement predictor. After all iterations are completed, only documents remaining in dataset C will be labeled by crowd workers. In the following subsections, we describe the details of our approach.

6.2.1 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 chapter: *Disagreement* and *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

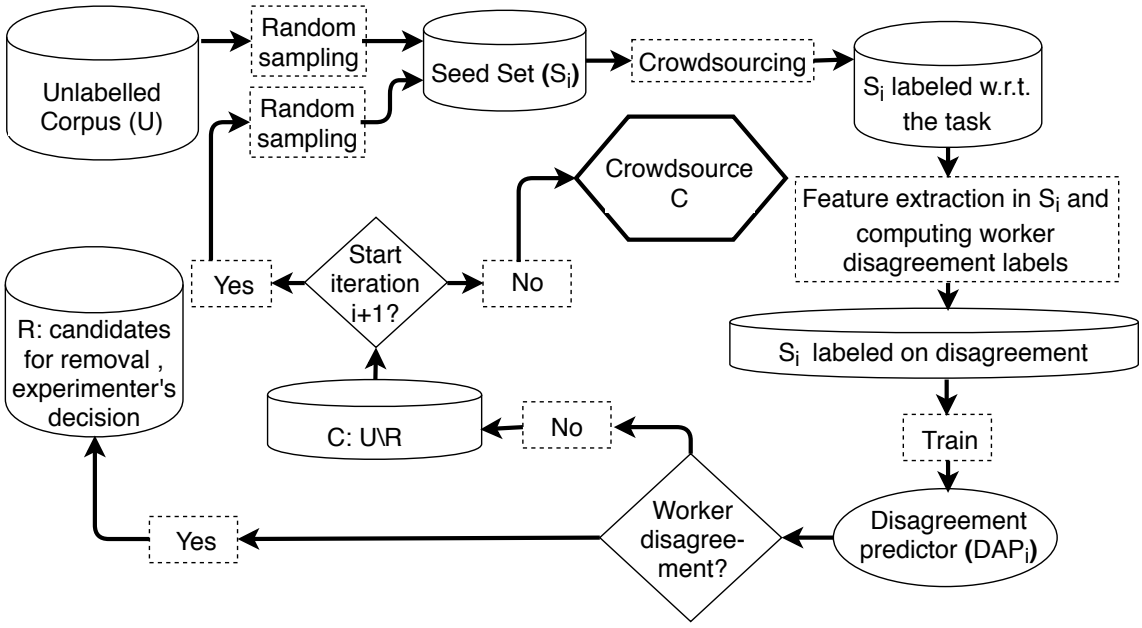


Figure 6.1: Schematic overview of our proposed methodology to obtain a more reliable dataset C for crowdsourcing, where i refers to the i^{th} iteration as described in the text.

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 non-ambiguous. Note that Definition 1 takes only worker disagreement into account as opposed Equation 5.1, which also considers labeling costs and predictor certainty as additional factors. We focus on worker disagreement because it is the most intuitive of the three factors following the argument "[crowd worker] disagreement is not noise, but signal" given by Aroyo et al. [33]. Moreover, we found evidence in our controlled experiment (cf. Section 3.1.4) that the crowd workers are faithful which lends more power to the previous argument. However, in the future we would also utilize the other two factors.

6.2.2 Disagreement Predictor

The disagreement predictor DAP_i plays an important role in our method as it reduces the size of the dataset to be labeled by the crowd by filtering out ambiguous documents. The initial seed set S_0 is created from the unlabeled dataset 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 (*Disagreement*, *No Disagreement*) 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 dataset R (*Disagreement*) or dataset C (*No Disagreement*) (line 14-17 in Algorithm 1). Therefore, C contains only the tweets $U \setminus (R \cup S_0)$, which are those document with predicted *No Disagreement*. 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 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 dataset 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 RQ3.1 (see Section 6.1) to test how well disagreement may be predicted.

Algorithm 1 Iteratively estimating the level of disagreement to remove ambiguous documents.

Input: Dataset of unlabeled documents (U).
Output: Set of documents to be labeled via crowdsourcing (C), set of ambiguous documents (R)
$S \leftarrow \emptyset$ ▷ seed set of previous iteration
$R \leftarrow \emptyset$
iteration i = 0;
repeat
    $C \leftarrow \emptyset$
    $N_i \leftarrow$ randSample($U \setminus S$, n) ▷ pick n documents
    crowdsource(N_i)
    $S_i \leftarrow$ createTrainingSet(N_i, S) ▷ see Algorithm 2
    DAP_i.train(S_i) ▷ train on disagreement labels
    for each document d in $U \setminus S_i$ **do**
        $label \leftarrow DAP_i$.predict(d)
        if $label$ == 'yes' **then**
            $R \leftarrow R \cup d$
        else
            $C \leftarrow C \cup d$
    $S \leftarrow S_i$
    $U \leftarrow C$ ▷ label propagation
    i = i + 1
until experimenter stops ▷ see section about the stopping criterion
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: $n \leftarrow$ allVotes(d) ▷ total votes
6: $m \leftarrow$ majVotes(d) ▷ #votes for majority label
7: $label \leftarrow$ 'no' ▷ *No Disagreement*
8: **if** $m \leq n/2$ **then**
9: $label \leftarrow$ 'yes' ▷ *Disagreement*
10: d.setDisagreement($label$)
11: **return** $N \cup S$

6.2.3 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 dataset 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 RQ3.2 (see Section 6.1).

6.3 Evaluation Framework

This section describes how we created a crowdsourced dataset 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*, *Neutral* (which corresponds to *Factual*). Irrelevant tweets are given the label *Irrelevant*.

6.3.1 The Dataset

We use TRAIN (cf. Section 3.1.1) as the seed set S_0. The dataset encompasses 500 tweets labeled hierarchically in terms of sentiment. Since the emerged trends in the dataset labeled by MD and SU are similar, we merge both datasets. This way, each of the 500 tweets received between 4-30 votes. The labeling procedure is described in Section 3.1.2. In addition, we utilize C (cf. Section 3.1.1), which contains 19.5k unlabeled tweets about the same topic. The main difference between both datasets is that tweets in C are shorter than those in TRAIN. To illustrate how these additional tweets look like, we present two tweets. On the first one from the dataset the crowd workers agreed:

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

On the second one below the workers disagreed:

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

6.3.2 Building Crowdsourced Datasets

To test how well our proposed methodology works, we want to collect a real-world crowdsourced dataset. At the same time, this crowdsourced dataset should be used to evaluate the performance of DAP_0 in practice. This leads to the following idea. To address the first requirement, we select unlabeled tweets from C and DAP_0 will be used to select these tweets. Instead of creating one crowdsourced dataset, we create three, namely one that contains only tweets with predicted LOW disagreement, one that comprises only tweets with predicted MEDIUM disagreement, and one that includes only tweets with predicted HIGH disagreement. Evaluating the disagreement levels allows us to draw conclusions about the performance of DAP_0, which was trained on S_0.

The detailed procedure for obtaining the three datasets to be crowdsourced is as follows. To obtain disagreement labels for all tweets in S_0, we estimate the worker disagreement DA in S_0 for tweet t by employing Equation 5.3. We then bin the resulting 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 predicts the worker disagreement in the remaining 19.5k tweets of C. To test the performance of DAP_0, we created three datasets - 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 Amazon Mechanical Turk for all three datasets 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 dataset to contain tweets with *Disagreement*, which we want to analyze. Building these three datasets allows us to analyze DAP_0's performance on real data in RQ3.2 (see Section 6.1). 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 Section 3.1.2 into a more suitable flat one using the labels *Positive*, *Negative*, *Neutral* (which corresponds to *Factual*) 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 *Disagreement* and tweets with *No Disagreement*. These two corrections allowed using the more intuitive majority voting scheme (see Definition 1) because Equation 5.3 does not yield continuous scores for a flat labeling scheme. In other words, Equation 5.3 was only used for creating the three datasets, but otherwise the flat scheme and binary worker disagreement labels were used throughout the chapter. The flat scheme was also applied to S_0 after the three datasets were created.

6.3.3 Features for Disagreement and Sentiment Classification

Table 6.1 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 chapter.

As shown in Table 6.1, 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) [64] which is a domain-independent lexicon and the SemEval-2015 English Twitter Lexicon (TWL) [65] which is specifically tailored to Twitter. In terms of sentiment, we also utilize subjective word lists proposed by [66]. 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. The reason for using these extended features is to account for mixed sentiment. Regarding the syntactic features, we obtain POS tags from Rosette[5] and NERs from Rosette and Watson.

[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

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 [60]
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 [61]
	F_{84}	Agreement AG (text, query) [62]
Topic	F_{85}-F_{94}	10 topics according to LDA
Word Embedding	F_{95}-F_{294}	Pre-trained Glove vectors [63]
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

Table 6.1: Overview of features used for sentiment and disagreement predictors.

Since there is a correlation between sarcastic tweets and worker disagreement [11], we include sarcasm-related features ($F_{59} - F_{67}$) as sarcasm increases ambiguity. On top of these, we generate ten topics from the whole dataset by using LDA [67], since topic features may also convey sarcasm-related information. Finally, we include word embeddings, specifically pre-trained Glove vectors [63] for Twitter[6], which may preserve semantic information.

Evaluating STP allows to investigate our core claim with RQ3.3 and RQ3.4 (see Section 6.1), namely that documents (here: tweets) affect predictor performance negatively and removing them might be helpful.

6.3.4 Label Distributions

For the classification experiments, it is necessary to consider the distribution of the sentiment labels which are shown in Figure 6.2 and Figure 6.3 respectively. In the former, four votes per tweet are used for the three crowdsourced datasets while all votes per tweet in S_0 are utilized. S_0 exhibits a similarly skewed label distribution as the three crowdsourced datasets, thus S_0 is representative. In all datasets similar patterns emerge in that the majority of tweets is considered *negative* while only a few tweets are *irrelevant*. Since the three crowdsourced datasets 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 Figure 6.3 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.

6.4 Results of Analysis

Before reporting the results of the four research questions, we first analyze how well our definition of worker disagreement matches reality because results based on an unsuitable

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

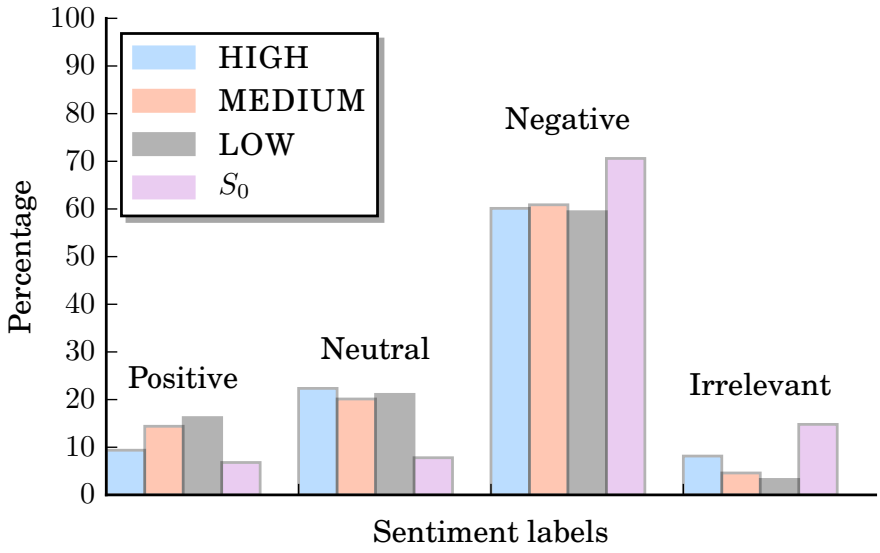


Figure 6.2: Label distribution across all four labeled datasets - three crowdsourced datasets using four votes per tweet and the seed set using all votes.

decision would render any findings meaningless. Afterwards we discuss all four research questions separately. We choose TRAIN as S_0 throughout all conducted experiments.

6.4.1 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 3.5k tweets, we identified four main sources that could induce 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”

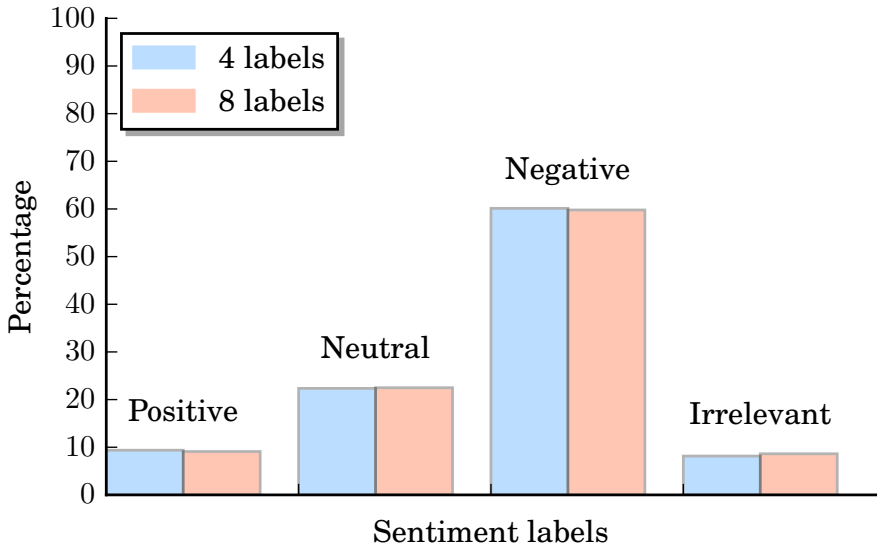


Figure 6.3: Label distribution in HIGH when computing majority labels using four and eight votes per tweet.

- 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

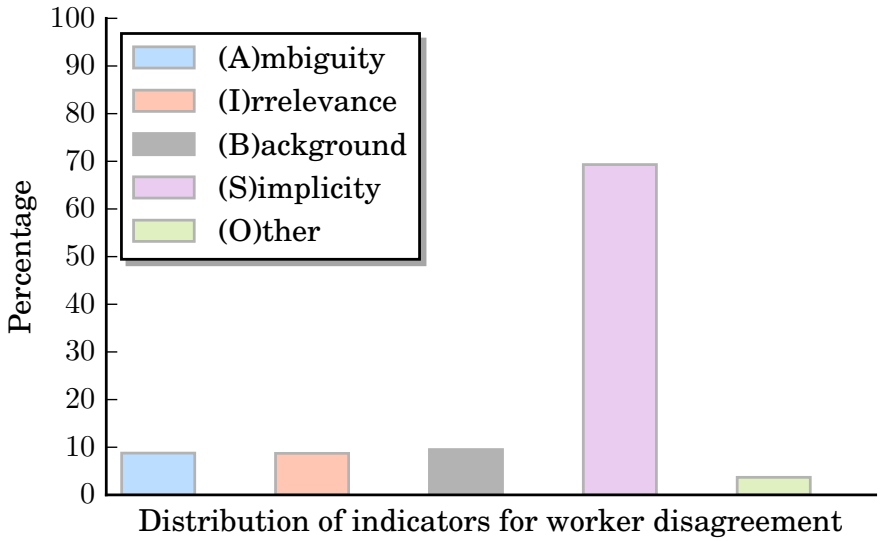


Figure 6.4: Distribution of the indicators inducing worker disagreement across 3.5k tweets.

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 Figure 6.4 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 non-ambiguous tweets. It turns out that 327/1106 ambiguous tweets according to Definition 1 are considered as non-ambiguous by the ground truth. One possible explanation for the differences could be that some crowd workers assigned low-quality labels. In terms of non-ambiguous tweets according to Definition 1, the ground truth considers 295/2394 non-ambiguous 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 non-ambiguous tweets sufficiently well.

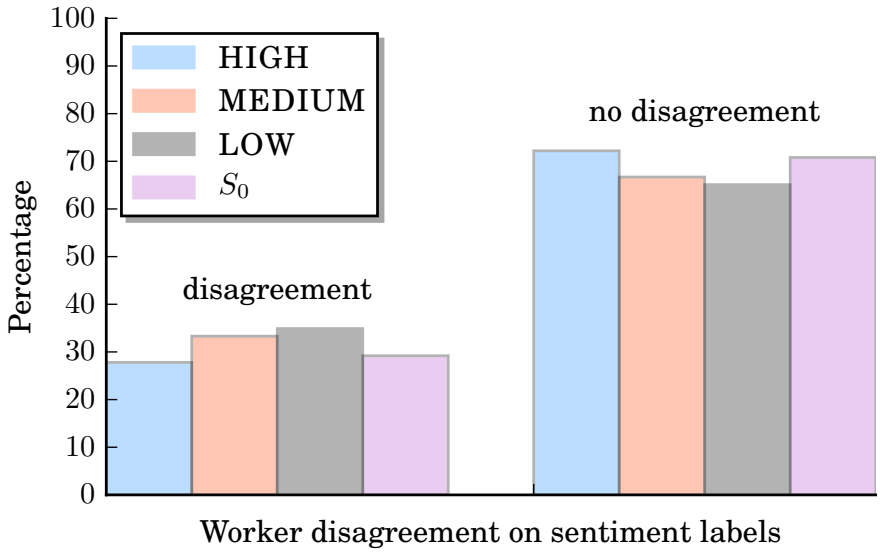


Figure 6.5: Worker disagreement distributions across all four labeled datasets - three crowdsourced datasets using four votes per tweet and the seed set using all votes.

6.4.2 Performance of the Disagreement Predictor

For analyzing RQ3.1, 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 Figure 6.5). DAP_0 separates ambiguous from non-ambiguous tweets. As dataset we employ S_0 and optimized DAP_0 for 15 min in Auto-Weka [68] 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 [69] with the optimized parameters reported by Auto-Weka. The results are shown in the first row of Table 6.2. The averaged AUC of 0.55 indicates that DAP_0 performs slightly better than chance which partially supports RQ3.1. However, the performance could be improved by tweaking the feature space which is beyond the scope of this chapter as we are mainly interested in general trends.

To analyze the performance of DAP_0 on unseen data, we computed the worker disagreement according to Definition 1 for each of the three crowdsourced datasets and illustrate the disagreement distribution in Figure 6.5. Four votes per tweet were used for the

three crowdsourced datasets as well as all votes per tweet in S_0. One would expect that the fraction of ambiguous tweets is highest in HIGH and lowest in LOW. However, it turns out that similar trends emerge in all datasets, namely workers disagree on around 30% of the tweets, which leads to a rejection of RQ3.1. 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, which is tested in the next section.

6.4.3 Gradual Improvement of the Disagreement Predictor

For our proposed method to work, the most important assumption is that DAP_i improves if S_i is expanded which is examined in RQ3.2. 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 RQ3.1 – *ambiguous* and *non-ambiguous*. As evaluation metric we utilize AUC and we train DAP_0 and DAP_1 as described in RQ3.1 using Auto-Weka. The results are shown in Table 6.2. An improvement in DAP_1 over DAP_0 of 6% supports RQ3.2 that our proposed methodology gradually refines the disagreement predictor over multiple iterations.

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

Table 6.2: AUC scores obtained in five Auto-Weka runs for DAP_0 trained on S_0 and DAP_1 trained on S_1 respectively.

Although the performance of DAP_1 shows room for improvement, e.g. by altering the feature space and identifying the most predictive features, an important remaining question is if this optimization is worth the effort? That is the reason why we run a simulation in the next section to assess the effect of difficult tweets on predictor performance

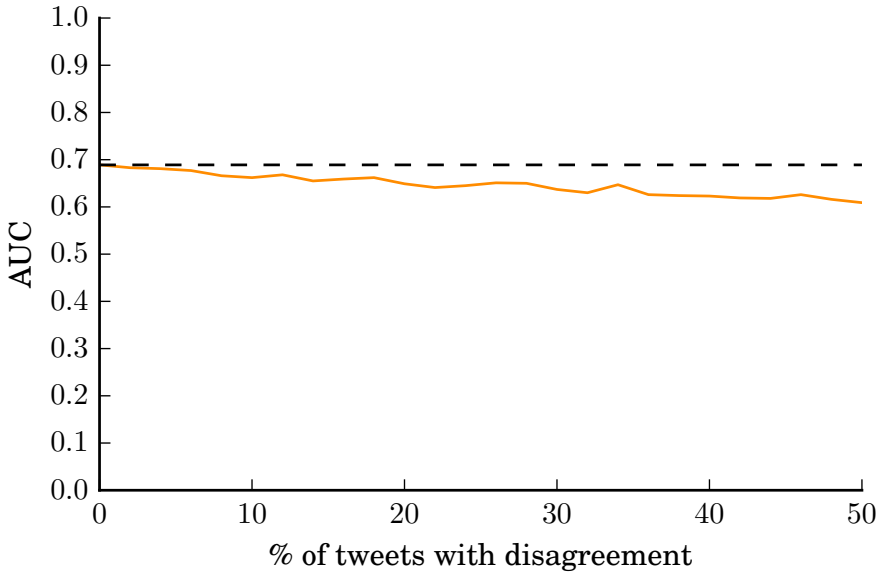


Figure 6.6: Influence of tweets with *Disagreement* on sentiment classification.

assuming that we can identify difficult tweets.

6.4.4 Effect of Ambiguous Tweets on Sentiment Classification

For analyzing RQ3.3, we devise the following simulation. We use S_1 from RQ3.2 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 datasets 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-

[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 dataset is comprised of up to 100% tweets with disagreement. Since

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 Figure 6.2). The reason for choosing random forest is that it is a predictor ensemble which tends to give more stable results than single predictors [70]. The result of our simulation is shown in Figure 6.6 and supports RQ3.3: 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%. One possible explanation for the performance drop of the predictor could be that difficult tweets were assigned a more or less random majority label due to worker disagreement. This majority label could then introduce noise in the respective set of documents that share the same label because existing patterns might be weakened or artificially introduced due to the random majority labels. Thus, discarding these difficult tweets from the training set could be a viable option. Another possible option for improved label reliability is requesting more labels for difficult tweets. This alternative strategy is evaluated in the next section by devising another simulation.

6.4.5 Effect of Allocating More Budget to Ambiguous Tweets on Sentiment Classification

To address RQ3.4, 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 Figure 6.7 in which we plotted the fraction of tweets with *Disagreement* over all three crowdsourced datasets considering only the first n labels, where $n = 2...8$. For $n = 2...4$ we computed the disagreement for each of the three datasets, while starting from $n = 5$ only HIGH is used because the other datasets received only four votes. The plot illustrates that the valleys and peaks start to converge when

this scenario is less realistic, we show the results only in Appendix B.

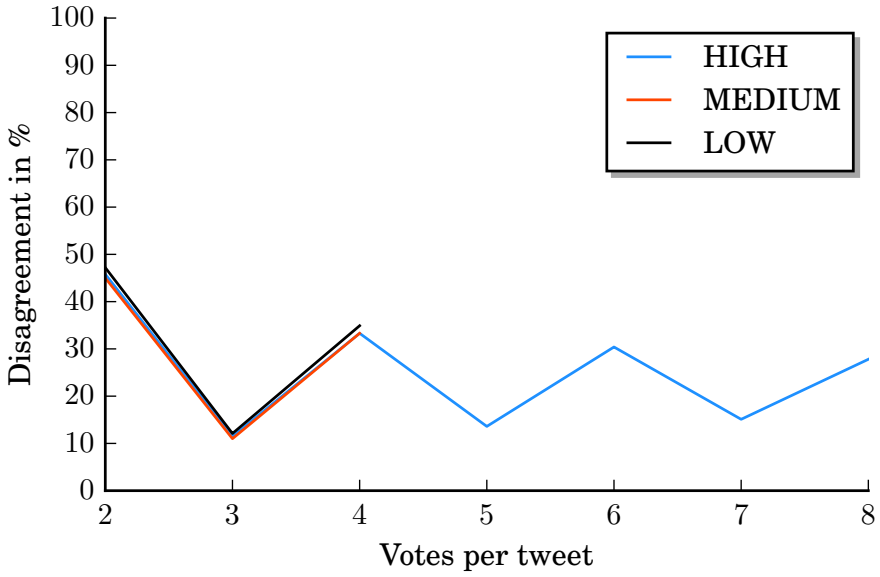


Figure 6.7: Fraction of tweets with *Disagreement* when using only the first n votes for deriving majority labels. For $n = 2, 3, 4$ we depict the fractions separately for LOW, MEDIUM, and HIGH, while for $n > 4$ only tweets from HIGH are available.

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.

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 RQ3.3 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 datasets. 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 dataset size to 174 tweets[8], initially all tweets are from ND

[8]Repeating the experiment with the same settings as in RQ3.3, now using only 87 tweets instead of 174

and then we gradually replace them by tweets from D in the same manner as in RQ3.3. The resulting performances of STP, for which we used again an optimized random forest predictor, are shown in Figure 6.8. They support RQ3.4 since the use of more votes does not improve the AUC scores. Surprisingly, contrary to RQ3.3, 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 RQ3.4 in that more votes do not improve AUC scores and in line with RQ3.3 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 dataset size and the randomized cross-validation splits because the other seven experiments in RQ3.3 and RQ3.4 using optimized and unoptimized predictors point to the opposite pattern in agreement with RQ3.3. Overall, our results support RQ3.4; 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 the presence of aleatoric uncertainty.

6.5 Discussion

In this chapter, 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, Figure 6.7 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

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. The results are depicted in Appendix C.

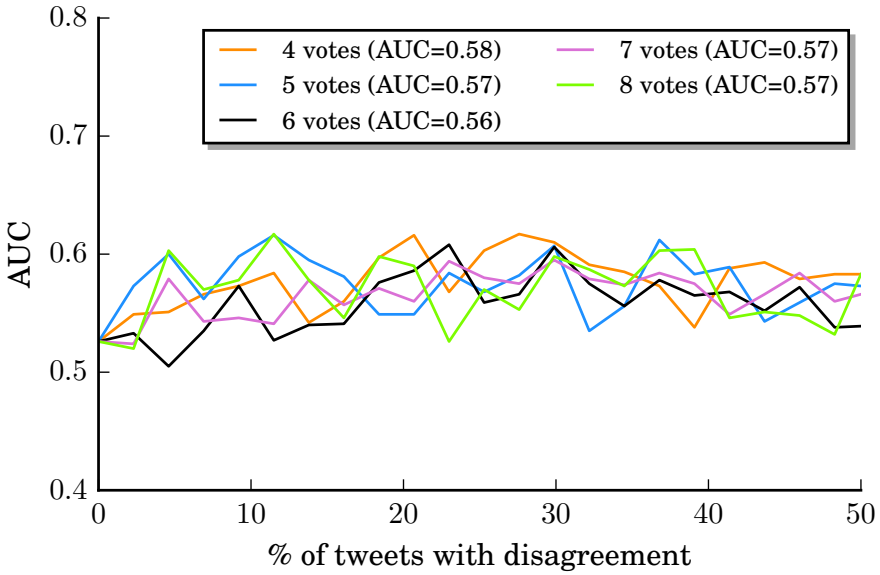


Figure 6.8: Influence of tweets with *Disagreement* on the predictor performance if the number of votes used for majority voting increases. The AUC scores in the legend are averaged per curve.

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 dataset, 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 tweets with *Disagreement* should be treated, while tweets with *No Disagreement* are given to the crowd workers. Nevertheless, we plan to experiment with different tweet representations like [71] to improve the performance of the disagreement predictor. Another potential avenue for identifying a better feature space for the disagree-

ment predictor is indirectly described in Section 6.4.1 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 if one requires reliable labels. Especially analyzing why some tweets are considered non-ambiguous 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 non-alleviatable 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.

A 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 shortcoming 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 6.7. 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 datasets are needed to understand when and how the disagreement may converge.

Our tweet dataset 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.

[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

Chapter 7

Conclusion and Future Work

This chapter first sums up the thesis and then discusses implications of our findings of our findings. Afterwards, future research directions that go beyond the topics discussed in Chapters 4-6 are outlined.

7.1 Summary

Although crowdsourcing is a popular mechanism to obtain large-scale labeled datasets for training supervised predictors, it is still problematic to obtain accurate and reliable labels despite the use of various human factors on crowdsourcing platforms. Therefore, we set out in this thesis to improve the reliability of crowdsourced datasets.

We started with an analysis of how crowd workers label documents. This was motivated by the idea that crowd workers undergo a learning process, which was measured in terms of annotation time, i.e. the time workers required to assign labels. We studied how this process evolved over time and how it affected the reliability of the labels that crowd workers assigned. To increase the validity of our resuls, we performed the crowdsourcing experiment in two different geographic locations independently – once in Sabancı, SU, (Turkey) and once in Magdeburg, MD, (Germany) – to distinguish local from potentially global patterns. However, most of the identified temporal patterns were similar in both locations indicating that the crowd workers were faithful and that the observed patterns

were not a coincidence. Most importantly, the results suggested that the learning process affects crowd workers' label reliability.

Since the effect of document difficulty on label reliability has not been investigated before, we examined this connection in a preliminary study using the dataset from our previous analysis. First, we derived difficulty labels for all documents before training sentiment predictors. The results suggest that difficult documents reduce label reliability.

Combining the findings of these two studies allowed us to propose a new crowdsourcing methodology that separates difficult from easy documents. To maximize label reliability, only easy documents are to be labeled by the crowd, while the remaining documents are to be handled differently depending on the task, e.g. by discarding them or labeling only a subset of them. The main advantage of our proposed methodology is that it is applied as a preprocessing step before crowdsourcing. Thus, it complements existing approaches that increase label reliability in a postprocessing step, e.g. by identifying spammers and removing their submitted labels.

7.2 General Conclusion

In Chapter 4 we found that label reliability increases in the exploitation phase. This emphasizes the importance of training crowd workers in advance so that they always label micro-tasks in this phase. Therefore, experimenters should provide crowd workers with a training session of appropriate length allowing them to reach their exploitation phase before they start labeling actual micro-tasks. Estimating the appropriate length of the learning phase depends on the labeling task, so experimenters should perform preliminary experiments to approximate the length accurately, for example by applying our methods from Sections 4.3.1 and 4.3.2. However, this works only if the crowdsourcing platform shares the annotation times of the crowd workers with the experimenter, which is usually the case, e.g. for Amazon Mechanical Turk and CrowdFlower. Otherwise the length of the learning phase must be estimated, e.g. by setting the length based on a similar labeling task for which the length of the learning phase is known. At the same time,

our finding motivates the necessity of retaining experienced crowd workers (i.e. those who have reached their exploitation phase) motivated for the current labeling task, so that they keep on completing related micro-tasks. Motivation can be increased to some extent by providing extrinsic incentives such as gradually increasing the rewards for completing more micro-tasks. However, it is known that higher monetary rewards do not necessarily lead to more reliable labels [72]. Thus, there is a trade-off between intrinsic and extrinsic motivation and a mixture of of both types of motivation seems more promising. A possible idea for intrinsic motivation would be allowing more experienced workers to rate the quality of the micro-tasks of inexperienced workers to build up their reputation as experts. An alternative to that strategy would be using a peer-dependent reward scheme [73], where crowd workers are paired up and exchange some personal information like their names and nationality. Either working as a team or competitively against each other, their motivation to submit high-quality labels increases leading to an improved label reliability.

When analyzing the interplay between document difficulty and label reliability, our preliminary study in Chapter 5 demonstrated that difficult documents reduced the label reliability in the exploitation phase of crowd workers. This motivates the idea that difficult documents should not necessarily be labeled by inexperienced crowd workers, but rather experts or not at all depending on the type of difficulty encountered in the documents – either aleatoric (a document is inherently ambiguous, i.e. additional labels will not converge to a majority label) or epistemic (a document is difficult, so requesting more labels could make the majority label converge) uncertainty.

Our proposed crowdsourcing methodology in Chapter 6 implements this idea of separating difficult documents from the rest, so that difficult documents are excluded from crowdsourcing to increase label reliability. The result obtained in Section 6.4.4 shows that excluding difficult documents from the training set improves the performance of the resulting predictor. This might be due to difficult tweets introducing noise that prevents the sentiment predictor from estimating the polarity more accurately. Our result from Section 6.4.5 offers an alternative explanation for this finding: many difficult tweets from

our dataset are inherently ambiguous therefore requesting more labels for difficult documents does not resolve crowd worker disagreement. This interpretation suggests that it is insufficient to separate difficult documents from the rest. Instead, one might have to increase the granularity of worker disagreement and distinguish between aleatoric and epistmic uncertainty among difficult documents.

7.3 Future Work

Besides raising many unanswered questions in Chapters 4-6, there are two additional avenues for future research that have been briefly mentioned in the previous section. First, our results suggest that removing difficult documents is a viable option and better than requesting more labels – but is this finding limited to our dataset or not? We found that many tweets are inherently difficult in our dataset, which would explain why removing them is more promising than requesting more labels for them. However, the question for future research is how common is aleatoric uncertainty in datasets? We suspect that the fraction of inherently difficult documents depends on the topic and labeling task. For example, longer text documents might be less likely to exhibit aleatoric uncertainty as opposed to short texts such as tweets. Similarly, the popularity of a topic could contribute to aleatoric uncertainty. If the fraction of epistemic uncertainty is higher in a dataset, we expect that requesting more labels for difficult documents will be superior to removing them.

This leads to the second research direction, namely training a predictor that teases apart aleatoric from epistemic uncertainty. Depending on which type of uncertainty is prevalent, a different strategy (removing a document vs. requesting more labels) could be utilized. However, before tackling this problem, one should improve the disagreement predictor (DAP), e.g. by exploring different feature spaces, because at the moment it is unclear how difficult it is to estimate the level of crowd worker disagreement. In this thesis we have implicitly assumed that it is easier to predict worker disagreement than the labels of the actual labeling task (here: sentiment analysis). Provided that our assumption

holds, the simulations in Sections 6.4.4 and 6.4.5 demonstrate the benefits of discarding difficult documents before crowdsourcing the rest given that a dataset contains inherently difficult documents.

Appendix A

Statistical Tests

In this section we list the statistical tests we use with their null hypotheses and underlying assumptions. If a test is used only for one specific experiment, we give specific examples from the main text for the variables involved in the computation of the tests. The term ”population”, which we use frequently in the explanations of the tests below, refers to the set of observations that are relevant for an experiment [74].

A.1 Wilcoxon Rank Sum Test

The Wilcoxon rank sum test is the non-parametric version of the unpaired student-t test, which compares the differences in two populations. All information in this section is taken from [75].

Null Hypothesis: The medians of the two populations are the same.

Assumptions:

1. The two samples are independent
2. The two populations have equal variance (homoscedastic)

A.2 ANOVA

It tests the difference between group means after any other variance in the dependent variable (here: median annotation time) is accounted for. The different sources of variance are called *levels*. All information in this section is taken from [76].

Null Hypothesis: The means of the dependent variable are the same for the different levels (here: *Learn*, *Rest*, *Fatigue*) of the given data.

Assumptions:

1. The samples are independent
2. The populations have equal variance (homoscedastic)
3. The populations are normally distributed
4. Each group in the ANOVA table should contain at least around 10 samples.

A.3 Fisher's Exact Test

This test is performed on a 2x2 contingency table with two nominal variables and the goal is to determine if the proportions of one variable are different depending on the value of the other variable. All information in this section is taken from [76].

Null hypothesis: the relative proportions of one variable are independent of the other variable.

Assumptions:

1. The samples are independent
2. Row and column totals of the contingency table are fixed

Appendix B

RQ3.3: Additional Results

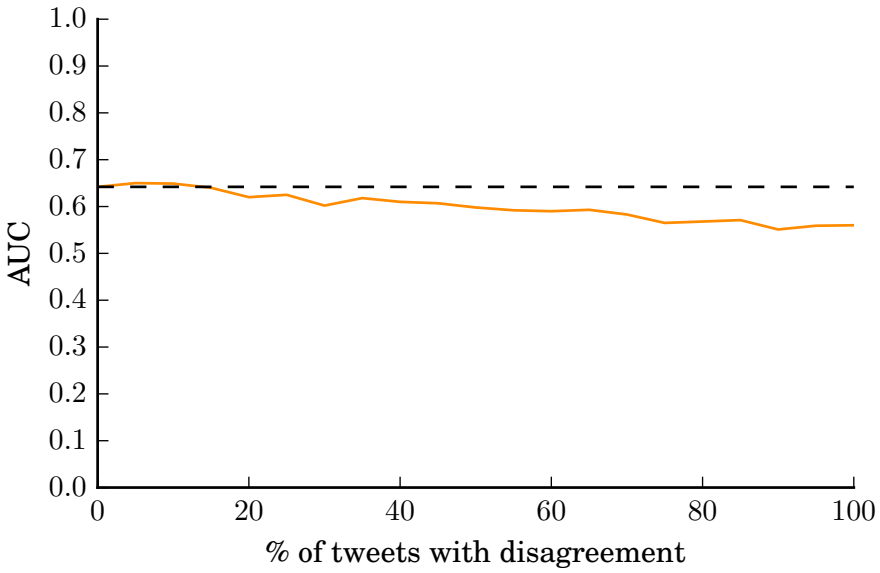


Figure B.1: Influence of tweets with *Disagreement* on sentiment classification using 1100 tweets for ND and D.

Appendix C

RQ3.4: Additional Results

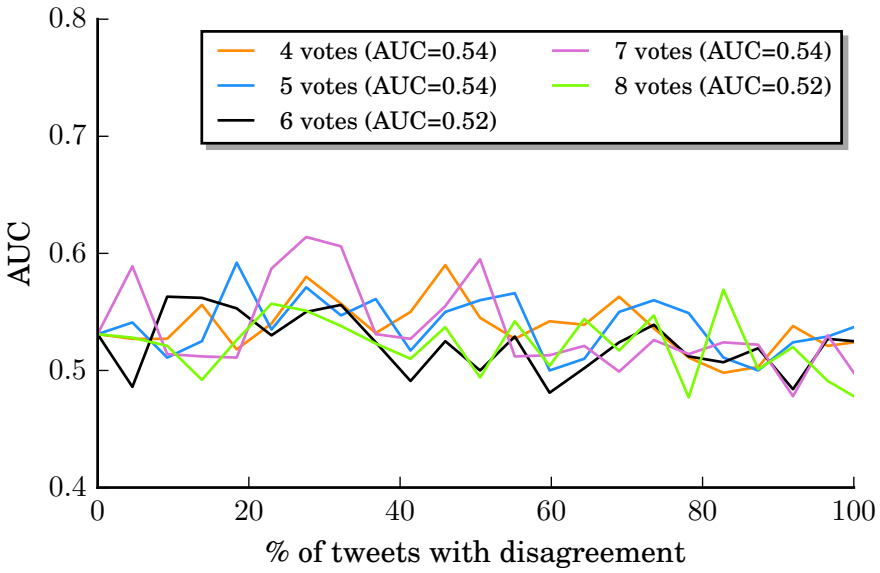


Figure C.1: Influence of tweets with *Disagreement* on the predictor performance if the number of votes used for majority voting increases. The AUC scores in the legend are averaged per curve. 87 tweets are used for ND and D.

Bibliography

[1] P. J. Oh, J. Chen, D. Hatcher, H. Djaladat, and A. J. Hung, "Crowdsourced versus expert evaluations of the vesico-urethral anastomosis in the robotic radical prostatectomy: is one superior at discriminating differences in automated performance metrics?," *Journal of Robotic Surgery*, Apr 2018.

[2] L. See, A. Comber, C. Salk, S. Fritz, M. van der Velde, C. Perger, C. Schill, I. McCallum, F. Kraxner, and M. Obersteiner, "Comparing the quality of crowdsourced data contributed by expert and non-experts," *PloS one*, vol. 8, no. 7, pp. 1–11, 2013.

[3] J. Lorenz, H. Rauhut, F. Schweitzer, and D. Helbing, "How social influence can undermine the wisdom of crowd effect," *Proceedings of the National Academy of Sciences*, vol. 108, no. 22, pp. 9020–9025, 2011.

[4] P. Donmez and J. G. Carbonell, "Proactive learning: cost-sensitive active learning with multiple imperfect oracles," in *Proceedings of the 17th ACM conference on Information and knowledge management*, pp. 619–628, ACM, 2008.

[5] F. Laws, C. Scheible, and H. Schütze, "Active learning with amazon mechanical turk," in *Proceedings of the conference on empirical methods in natural language processing*, pp. 1546–1556, Association for Computational Linguistics, 2011.

[6] Y. E. Kara, G. Genc, O. Aran, and L. Akarun, "Modeling annotator behaviors for crowd labeling," *Neurocomputing*, vol. 160, pp. 141–156, 2015.

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

[8] 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.

[9] T. Itoko, S. Arita, M. Kobayashi, and H. Takagi, "Involving senior workers in crowdsourced proofreading," in *International Conference on Universal Access in Human-Computer Interaction*, pp. 106–117, Springer, 2014.

[10] 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*, pp. 5–14, ACM, 2017.

[11] 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." Poster presented at: 4th AAAI Conference on Human Computation and Crowdsourcing (HCOMP); Oct 30 - Nov 3; Austin, TX, 2016.

[12] U. Gadiraju, R. Kawase, and S. Dietze, "A taxonomy of microtasks on the web," in *Proceedings of the 25th ACM conference on Hypertext and social media*, pp. 218–223, ACM, 2014.

[13] B. Settles, M. Craven, and L. Friedland, "Active learning with real annotation costs," in *Proceedings of the NIPS workshop on cost-sensitive learning*, pp. 1–10, 2008.

[14] A. I. Chittilappilly, L. Chen, and S. Amer-Yahia, "A survey of general-purpose crowdsourcing techniques," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 9, pp. 2246–2266, 2016.

[15] B. Settles, "Active learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 6, no. 1, pp. 1–114, 2012.

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

[17] C.-y. Lee and J. Glass, "A transcription task for crowdsourcing with automatic quality control," in *Twelfth Annual Conference of the International Speech Communication Association*, vol. 11, Citeseer, 2011.

[18] 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)*, pp. 197–205, 2017.

[19] 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)*, pp. 139–148, 2016.

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

[21] A. Calma, J. M. Leimeister, P. Lukowicz, S. Oeste-Reiß, T. Reitmaier, A. Schmidt, B. Sick, G. Stumme, and K. A. Zweig, "From active learning to dedicated collaborative interactive learning," in *ARCS 2016; 29th International Conference on Architecture of Computing Systems; Proceedings of*, pp. 1–8, VDE, 2016.

[22] P. Dai, C. H. Lin, D. S. Weld, *et al.*, "Pomdp-based control of workflows for crowdsourcing," *Artificial Intelligence*, vol. 202, pp. 52–85, 2013.

[23] J. M. Rzeszotarski and A. Kittur, "Instrumenting the crowd: using implicit behavioral measures to predict task performance," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pp. 13–22, ACM, 2011.

[24] S. Zhu, S. Kane, J. Feng, and A. Sears, "A crowdsourcing quality control model for tasks distributed in parallel," in *CHI'12 Extended Abstracts on Human Factors in Computing Systems*, pp. 2501–2506, ACM, 2012.

[25] G. Kazai and I. Zitouni, "Quality management in crowdsourcing using gold judges behavior," in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pp. 267–276, ACM, 2016.

[26] S. Han, P. Dai, P. Paritosh, and D. Huynh, "Crowdsourcing human annotation on web page structure: Infrastructure design and behavior-based quality control," *ACM Trans. Intell. Syst. Technol.*, vol. 7, pp. 56:1–56:25, Apr. 2016.

[27] U. Gadiraju, R. Kawase, S. Dietze, and G. Demartini, "Understanding malicious behavior in crowdsourcing platforms: The case of online surveys," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 1631–1640, ACM, 2015.

[28] H.-Â. Cao, F. Rauchenstein, T. K. Wijaya, K. Aberer, and N. Nunes, "Leveraging user expertise in collaborative systems for annotating energy datasets," in *Big Data (Big Data), 2016 IEEE International Conference on*, pp. 3087–3096, IEEE, 2016.

[29] 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*, pp. 100–114, Springer, 2015.

[30] J. Baldridge and A. Palmer, "How well does active learning actually work?: Time-based evaluation of cost-reduction strategies for language documentation," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pp. 296–305, Association for Computational Linguistics, 2009.

[31] S. Arora, E. Nyberg, and C. P. Rosé, "Estimating annotation cost for active learning in a multi-annotator environment," in *Proceedings of the NAACL HLT 2009 Work-*

shop on Active Learning for Natural Language Processing, pp. 18–26, Association for Computational Linguistics, 2009.

[32] E. K. Ringger, M. Carmen, R. Haertel, K. D. Seppi, D. Lonsdale, P. McClanahan, J. L. Carroll, and N. Ellison, "Assessing the costs of machine-assisted corpus annotation through a user study.," in *LREC*, vol. 8, pp. 3318–3324, 2008.

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

[34] 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)*, p. 89, 2015.

[35] 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.

[36] 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*, pp. 1345–1354, ACM, 2015.

[37] 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*, pp. 2623–2634, ACM, 2016.

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

[39] M. Martinez-Alvarez, A. Bellogin, and T. Roelleke, "Document difficulty framework for semi-automatic text classification," in *International Conference on Data Warehousing and Knowledge Discovery*, pp. 110–121, Springer, 2013.

[40] A. Culotta and A. McCallum, "Reducing labeling effort for structured prediction tasks," in *AAAI*, vol. 5, pp. 746–751, 2005.

[41] 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.

[42] M.-S. Paukkeri, M. Ollikainen, and T. Honkela, "Assessing user-specific difficulty of documents," *Information Processing & Management*, vol. 49, no. 1, pp. 198–212, 2013.

[43] D. Carmel, E. Yom-Tov, A. Darlow, and D. Pelleg, "What makes a query difficult?," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 390–397, ACM, 2006.

[44] 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.), pp. 2035–2043, Curran Associates, Inc., 2009.

[45] J. Goncalves, M. Feldman, S. Hu, V. Kostakos, and A. Bernstein, "Task routing and assignment in crowdsourcing based on cognitive abilities," in *Proceedings of the 26th International Conference on World Wide Web Companion*, pp. 1023–1031, International World Wide Web Conferences Steering Committee, 2017.

[46] S. Räbiger, M. Spiliopoulou, 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.

[47] D. Feng, S. Besana, K. Boydston, and G. Christian, "Towards high-quality data extraction via crowdsourcing," in *In Proceedings of the The World's First Conference on the Future of Distributed Work (CrowdConf-2010*, 2010.

[48] S. Kiritchenko, S. Matwin, R. Nock, and A. F. Famili, "Learning and evaluation in the presence of class hierarchies: Application to text categorization," in *Conference of the Canadian Society for Computational Studies of Intelligence*, pp. 395–406, Springer, 2006.

[49] C. N. Silla Jr and A. A. Freitas, "A survey of hierarchical classification across different application domains," *Data Mining and Knowledge Discovery*, vol. 22, no. 1-2, pp. 31–72, 2011.

[50] D. Kahneman and A. Tversky, "On the psychology of prediction.," *Psychological review*, vol. 80, no. 4, pp. 237–251, 1973.

[51] D. Kahneman and A. Tversky, "Prospect theory: An analysis of decision under risk," *Econometrica: Journal of the econometric society*, pp. 263–291, 1979.

[52] S. Räbiger, Y. Saygın, and M. Spiliopoulou, "How does tweet difficulty affect labeling performance of annotators?," tech. rep., Sabancı University, July 2018. `https://arxiv.org/pdf/1808.00388.pdf`.

[53] S. Yang, P. Dessai, M. Verma, and M. Gerla, "Freeloc: Calibration-free crowdsourced indoor localization," in *INFOCOM, 2013 Proceedings IEEE*, pp. 2481–2489, IEEE, 2013.

[54] O. Alonso and S. Mizzaro, "Can we get rid of trec assessors? using mechanical turk for relevance assessment," in *Proceedings of the SIGIR 2009 Workshop on the Future of IR Evaluation*, vol. 15, p. 16, 2009.

[55] P. Salomoni, C. Prandi, M. Roccetti, V. Nisi, and N. J. Nunes, "Crowdsourcing urban accessibility:: Some preliminary experiences with results," in *Proceedings of the 11th Biannual Conference on Italian SIGCHI Chapter*, pp. 130–133, ACM, 2015.

[56] E. Maddalena, M. Basaldella, D. De Nart, D. Degl'Innocenti, S. Mizzaro, and G. Demartini, "Crowdsourcing relevance assessments: The unexpected benefits of limit-

ing the time to judge," in *Fourth AAAI Conference on Human Computation and Crowdsourcing*, 2016.

[57] A. Kolobov, D. S. Weld, *et al.*, "Joint crowdsourcing of multiple tasks," in *First AAAI Conference on Human Computation and Crowdsourcing*, pp. 36–37, 2013.

[58] S. Räbiger, G. Gezici, M. Spliliopoulou, and Y. Saygın, "Predicting worker disagreement for more effective crowd labeling," in *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, IEEE, 2018.

[59] 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.

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

[61] 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*, pp. 331–340, ACM, 2010.

[62] 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*, pp. 2345–2350, ACM, 2013.

[63] 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)*, pp. 1532–1543, 2014.

[64] 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.

[65] 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), pp. 451–463, Association for Computational Linguistics, June 2015.

[66] 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), pp. 168–177, ACM, 2004.

[67] 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.

[68] 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.

[69] 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.

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

[71] 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*, pp. 1041–1044, ACM, 2016.

[72] W. Mason and D. J. Watts, "Financial incentives and the performance of crowds," *ACM SigKDD Explorations Newsletter*, vol. 11, no. 2, pp. 100–108, 2010.

[73] S.-W. Huang and W.-T. Fu, "Don't hide in the crowd!: increasing social transparency between peer workers improves crowdsourcing outcomes," in *Proceedings of the*

SIGCHI Conference on Human Factors in Computing Systems, pp. 621–630, ACM, 2013.

[74] J. F. Kenney, *Mathematics of statistics*. D. Van Nostrand Company Inc; Toronto; Princeton; New Jersey; London; New York,; Affiliated East-West Press Pvt-Ltd; New Delhi, 2013.

[75] E. L. Lehmann and H. J. D'Abrera, *Nonparametrics: statistical methods based on ranks*. Holden-Day, 1975.

[76] J. H. McDonald, *Handbook of biological statistics*, vol. 2. sparky house publishing Baltimore, MD, 2009.