Measuring Task Complexity in Human Computation Systems

Master's Thesis

Friso Cornelis Albertus Abcouwer

Measuring Task Complexity in Human Computation Systems

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE
TRACK INFORMATION ARCHITECTURE

by

Friso Cornelis Albertus Abcouwer
born in Leiderdorp, the Netherlands

Web Information Systems
Department of Software Technology
Faculty EEMCS, Delft University of Technology
Delft, the Netherlands
`http://wis.ewi.tudelft.nl`

© 2015 Friso Cornelis Albertus Abcouwer.

Measuring Task Complexity in Human Computation Systems

Author: Friso Cornelis Albertus Abcouwer
Student id: 4019873
Email: `f.c.a.abcouwer@student.tudelft.nl`

Abstract

Human computation platforms offer *requesters* the possibility to outsource *human intelligence tasks* or *HITs* to large amounts of *workers* around the world. Researching the various aspects of HIT design has the potential to improve the human computation experience for both requesters and workers. This thesis researches the relationship between quantitative aspects of HITs and their perceived complexity. Specifically, we consider three categories of task attributes: Metadata, Content and Visual. In order to measure these attributes, we constructed a real-time human computation market crawler that we used to gather HIT data and reinstantiate HITs from the largest human computation market, Amazon Mechanical Turk, and a task attribute analysis pipeline to obtain task attribute measurements from the market data. We present an analysis of the various task attributes, and perform an experiment predicting HIT throughput based on task attribute and market data. Our analysis shows that market statistics, task batch size and semantic content features can be used to predict throughput. We also conduct an initial exploration into measuring subjective HIT complexity as perceived by workers. Based on our analysis and experimental results, we identify three key points of advice for human computation task designers.

Thesis Committee:

Chair:	Prof. dr. ir. G.J. Houben, Faculty EEMCS, TUDelft
University Supervisor:	Dr. A. Bozzon, Faculty EEMCS, TUDelft
Committee Member:	Dr. M.A. Larson, Faculty EEMCS, TUDelft

Preface

This report is the final result of my Master's thesis project, marking the end of my time as a Computer Science student at Delft University of Technology after six amazing years. While the project had its ups and downs and was challenging from start to finish, I am proud of what I was able to achieve. Before I leave university for the 'real world', I would like to use this opportunity to thank several people.

To start with, I would like to thank Alessandro Bozzon, my supervisor during this project, whose guidance and feedback always pointed me in the right direction (even when I did not realize it right away) and motivated me to always think just one step further. I would also like to thank Geert-Jan Houben and Martha Larson, the other two members of my thesis committee, for their efforts and feedback. I also thank my mother and Max de Groot for proofreading early drafts of my thesis. A special thank you goes out to Jie Yang, whose support, insight and unfailing willingness to provide help when I needed it were incredibly useful.

Apart from the people I worked with during my thesis project, I would like to thank all of the friends I have made over the course of my university career. I feel truly privileged for having been able to experience and learn so much and meet so many wonderful people in the past six years.
Finally, I would like to thank my parents for supporting me in everything I do and always believing in me.

Friso Abcouwer
Delft, the Netherlands
October 21, 2015

Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction

In recent years, exponential increases in easily available computing power have enabled computer systems to complete tasks that used to be considered impossible or would have taken days, months or even years to complete. This increase in computation speed, however, has not led to computers being able to perform complex tasks such as those involving subjective judgements or natural language processing. While progress is being made in finding fully automated solutions for a lot of these types of challenges, for a very large amount of problems, state of the art automated solutions do not come close to providing a solution that can rival the performance of an average human working on the same task. However, work done by humans is usually significantly more costly and time-consuming, and the quality of the results can also be unpredictable.

A compromise in solving this dilemma can be found in the field of human computation. By outsourcing those tasks that can be completed more easily by one or more humans than by itself, an automated system can offer solutions to problems it would normally not be able to solve on its own, while still offering advantages inherent to an automated system like reliability, speed and low cost. We will refer to tasks completed by humans on such platforms as **Human Intelligence Tasks**, or **HITs** for short. There are many different types of human computation or crowdsourcing platforms, which we will elaborate on in more detail in Chapter 2.

Due to the incorporation of human elements as a part system functionality, designing a human computation platform is extremely challenging: not only do designers have to create a system that performs as it is required, but they have to manage the expectations and requirements of all parties involved on both sides of the market as well. To illustrate, we consider the market dynamics of Amazon Mechanical Turk. At one end, there are the requesters, who want their tasks to be completed as quickly and cheaply as possible with a result of the highest possible quality. Apart from investing in their task design, managing budgets and results and communicating with and rewarding their workers, requesters constantly have to be wary of malicious workers who input bogus answers to quickly earn money. At the other end, there are the workers, whose position as ‘cogs in the machine’ raises significant ethical concerns. For instance, many workers are dependent on the income they make from working on

MTurk to support themselves. Workers are also often treated unfairly by requesters, for example by having work rejected without being given a reason (and thus not receiving a reward for that work), and they do not enjoy employment protections that apply to 'regular' workers [34]. Some have gone as far as to say that using MTurk is 'exploitative' [19].

The situation described above exemplifies some important issues that need to be dealt with in order to advance the field of human computation. In short, the motivation for this thesis is the goal of progressing towards an ideal human computation platform that is fair and open towards workers while allowing requesters to maximize their results. There is certainly a lot of room for improvement until this type of platform becomes a reality. This thesis aims to help the human computation community take a step into that direction by investigating the way HITs are designed and presented to workers: it is our view that the way a HIT is constructed influences workers' motivation to take on the work, their performance during the HIT and their satisfaction afterward, all of which, when correctly managed, can improve the outcomes of crowdsourcing for both requesters and workers. The specific aspect of the crowdsourcing process this thesis focuses on is **task complexity**, which we will define as follows in Chapter 2:

The **complexity** of a HIT can be approximated by the sum of the complexity of its objectively measurable attributes, as perceived by its workers.

Our approach to evaluating HIT attributes is by linking them to throughput, which we argue can be a proxy for complexity, and by investigating the various attributes are interrelated. Our specific scope for this thesis is on the requester side of human computation. We wish to learn how we can use complexity as a means to improve the human computation experience for requesters. As a proxy for task complexity, we will investigate task throughput, and perform experiments to predict throughput based on task attributes. This should lead to insights allowing task requesters to design tasks with higher throughput than before. In the remainder of this chapter we will explain our methodology for measuring task complexity by linking this goal to our research objectives and how each one is represented in the thesis.

1.2 Research Objectives

Our research objectives can be summarized in the forms of research questions, to be answered based on the results of our work.

RQ1 *How can we measure the complexity of a human computation task?*

There are many different aspects by which a HIT can be evaluated. For example, one can look at the phrasing of instructions, the content, complexity and structure of the code that makes up the HIT page, the presence or absence of certain 'gold standard' design aspects, or how the worker perceives the complexity of the task. By studying historical data from MTurk, we will try to find which, if any, of the many attributes of a task can be indicative of complexity. In Chapter 2, we present a background on human computation and task complexity, as well as related work researching human intelligence task design. In Chapter 3, we identify key task attributes that we consider possible contributors to task complexity, and in Chapter 4, we describe two systems we constructed to support the measurement of HIT complexity.

RQ2 *What are the effects of the complexity of a HIT on its throughput?*

We are interested in identifying what effects the various factors that contribute to a task's complexity have on workers, specifically the job performance of workers. In Chapter 3, we argue that the throughput of a HIT can be used as a proxy for task complexity, and in Chapter 5, we evaluate historical data from Amazon Mechanical Turk to attempt to find a relationship between task complexity and throughput.

RQ3 *Can we predict the complexity of a human computation task based on its attributes?*

Finding a way to predict complexity can tell us more about the relationship between factors that contribute to a task's complexity and the task itself. Furthermore, if we can identify some attributes that correlate with how complex a task is, it might be possible to construct some method to apply this knowledge, leading to **RQ4**. In Chapter 5, we present an experiment predicting HIT throughput based on task attribute data.

RQ4 *How can the concept of task complexity be used to support human computation task design?*

If we find some way to measure task complexity and have an idea of its effects, we can offer recommendations that task designers can incorporate to improve their results. In Chapter 5, we identify several key recommendations for HIT designers based on our results.

1.3 Contributions

To answer our research questions, we first need to *gather HIT data* and *measure HIT attributes*. Then, we wish to *analyze HIT content* to find a possible link between task attributes and objective HIT complexity. Finally, we wish to *predict HIT throughput*. Additionally, using gathered HIT data, we can *reinstantiate HITs* and *measure subjective HIT complexity*. Based on these steps, we identify the scientific contributions of this work as follows:

1. **HIT Crawling Platform:** We constructed a real-time human computation market tracker for the largest human computation market, Amazon Mechanical Turk. This system, based on the existing MTurk Tracker platform, crawls the MTurk market in real time and has additional features for storing the content of HITs. This allows for HITs to be restored to be performed again at a later point in time with a greater degree of accuracy than was possible in the past, which opens up an avenue of research opportunities. The architecture of this platform is described in Chapter 3, and we identify possible future uses for it in Chapter 6.

2. **Analysis of HIT Content:** Based on literature, we define three categories of attributes for human intelligence tasks: Metadata, Content and Visual. Then, using historical market data from the largest human computation platform, Amazon Mechanical Turk, we perform a study to try and find relationships between the quantitative properties of HITs and task throughput, which we argue can be used as a proxy for task complexity. Based on this study, we identify how task attributes can be used to measure task complexity and link this to possible new 'best practices' in HIT creation. This analysis is provided in Chapter 5 and our conclusions can be found in Chapter 6.

3. **Throughput Prediction:** Using the results from our study, we re-enact the throughput prediction experiment by Difallah et al. [11] to investigate if the attributes we identified can be used to predict throughput. We describe this experiment in Chapter 5.

4. **Reinstantiated Human Intelligence Task Dataset:** Having tracked and stored a large amount of HITs from the AMT market, we manually filtered and curated this data, constructing a dataset of 62 diverse tasks, all from different requesters, that have been completely reinstantiated. We used this dataset in an initial experiment to measure subjective task complexity, which is presented in Chapter 5. In Chapter 6, we discuss the potential of further research using reinstantiated HITs.

1.4 Thesis Outline

The remainder of this thesis is organised as follows. Chapter 2 introduces the scientific background of this thesis and discusses related work. Specifically, it will introduce the concepts of crowdsourcing and human computation as well as the context of our study, Amazon Mechanical Turk. Then, we will examine previous research into the various aspects of HIT design and task complexity. Chapter 3 describes our methodology: we outline the task attributes we chose to incorporate in our study and our motivations for each one. We also describe our method for evaluating complexity and motivate our decision to use HIT throughput as an approximation of complexity. Chapter 4 describes the technical means we used to achieve our results. First, we describe the MTurk Tracker platform and our extensions to it, giving an overview of how the data was retrieved and formatted. Then, we explain our HIT attribute analysis pipeline, which we used to retrieve attribute measurements from the data. Chapter 5 describes our results: we elaborate on what we learned from studying the obtained data, and evaluate experimental results for throughput prediction on two distinct datasets. We describe an initial experiment to measure subjective complexity, and offer guidelines to human computation task designers based on our results. Finally, in Chapter 6, we will conclude our work by summarizing the insights we gained from the study and identifying possible directions for future research.

Chapter 2

Background and Related Work

In this chapter, previous work in the area of human computation will be introduced to provide the scientific context for our research and to clearly define important concepts based on literature. First, we will give an introduction to the field of human computation, specifically its history and its existing applications. Then, we give an overview of previous research into Mechanical Turk focused on both the requester and worker sides of the crowdsourcing process. Finally, we present an overview of previous research into various aspects of task design and based on this motivate our decision to focus on a single aspect: task complexity.

2.1 Introduction to Human Computation

2.1.1 Human Computation

An overview of the history of human computation is given by Quinn and Bederson's 2011 survey [50]. The term 'human computation' was used as early as the 19th century, but current research, including this thesis, is focused on the modern usage of the term. Quinn and Bederson refer to the following definition given by Von Ahn in 2005 [59]: " *...a paradigm for utilizing human processing power to solve problems that computers cannot yet solve"*. Von Ahn, along with co-author Law [41], later defined human computation as being *computation*, which is defined as " *the process of mapping of some input representation to some output representation using an explicit, finite set of instructions"*, carried out by a human. Combining Von Ahn and Law's definitions with Quinn and Bederson's own observations, we propose the following definition to be considered for the purposes of this thesis:

Human Computation *is the utilization by a computation system or process of human processing power to perform its function.*

We also use Law and Von Ahn's definition of human computation systems: *"intelligent systems that organize humans to carry out the process of computation - whether it be performing the basic operations (or units of computation), taking charge of the control process itself (e.g., decide what operations to execute next or when to halt the program), or even synthesizing the program itself (e.g., by creating new operations and specifying how they are ordered)."*

2.1.2 Crowdsourcing

One of the most popular applications of the human computation paradigm has been its use in crowdsourcing. The term crowdsourcing was coined by Jeff Howe in a 2006 Wired article [28]. He defines it on his website as follows: *"Crowdsourcing is the act of taking a job traditionally performed by a designated agent (usually an employee) and outsourcing it to an undefined, generally large group of people in the form of an open call."*[1].

In 2012, Estellés-Arolas and González-Ladrón-de-Guevara proposed a more extensive definition based on a literature survey into the various defining characteristics of the crowdsourcing process. Their definition was as follows:

"Crowdsourcing is a type of participative online activity in which an individual, an institution, a non-profit organization, or company proposes to a group of individuals of varying knowledge, heterogeneity, and number, via a flexible open call, the voluntary undertaking of a task. The undertaking of the task, of variable complexity and modularity, and in which the crowd should participate bringing their work, money, knowledge and/or experience, always entails mutual benefit. The user will receive the satisfaction of a given type of need, be it economic, social recognition, self-esteem, or the development of individual skills, while the crowdsourcer will obtain and utilize to their advantage that what the user has brought to the venture, whose form will depend on the type of activity undertaken." [17]

In addition to being more detailed than Howe's original definition, this definition also notably adds the requirement that the process take place online - crowdsourcing is, according to the authors, necessarily a *"participative distributed online process"*. While an 'offline' form of crowdsourcing might be conceivable, Estellés-Arolas and González-Ladrón-de-Guevara found that none of the work they surveyed made mention of this and that some authors argue that the web provides the technological basis required to perform crowdsourcing. For these reasons, this thesis will also only consider online crowdsourcing.

Hosseini et al. [27] identify four main pillars of crowdsourcing: the crowdsourcer, the crowd, the crowdsourced task and the crowdsourcing platform. Because of the varying perspectives on the concept of crowdsourcing from different fields, they argue that coming up with a common definition for crowdsourcing might be unfeasible. For our own purposes, however, we do require a single definition of crowdsourcing in order to define the context of our research in an unambiguous way. Based on the works cited above, we propose the following definition of crowdsourcing to be used in the remainder of this thesis:

***Crowdsourcing** is the act in which the crowdsourcing party solicits, through an open online call, the completion of a certain task from a crowd of individual workers in exchange for a reward.*

We then define a **crowdsourcing system** or **crowdsourcing platform** as an online system that facilitates the operation and administration of the crowdsourcing process.

[1]`http://crowdsourcing.typepad.com/`

2.1.3 Integrating Crowdsourcing and Human Computation

Crowdsourcing platforms are an example of the human computation paradigm being used to facilitate the crowdsourcing process. These platforms are computation systems that allow crowdsourcers to place an open call asking a crowd of workers to complete certain tasks.

As Quinn and Bederson put it, *"The intersection of crowdsourcing with human computation ... represents applications that could reasonably be considered as replacements for either traditional human roles or computer roles."* In his Master's thesis, Debarshi Basak gives an overview of the different types of human computation systems available online [3]. He cites Doan et al. [13], who give an overview of a large amount of modern online crowdsourcing systems, making a distinction between crowdsourcing systems that require explicit or implicit worker participation. Examples of systems with implicit participation are games with a purpose like the ESP game [60] or reCAPTCHA [61], both introduced by Von Ahn, and examples of systems with explicit worker participation are Amazon Mechanical Turk and Crowdflower[2]. Basak also notes the distinction made by Hirth et al. [26] regarding platforms specialized in a certain kind of task, like DuoLingo[3], and non-specialized generalistic platforms like MTurk. In addition to the platforms mentioned above, there are a great deal of other interesting applications of crowdsourcing that are beyond the scope of this thesis, such as citizen science [47], medical applications [15], and many more. For further reading, we recommend Yuen et al. [65] and Kittur et al. [39] and their cited works.

2.2 Crowdsourcing on Amazon Mechanical Turk

This thesis focuses on the most popular crowdsourcing platform, Amazon Mechanical Turk or *MTurk*[4]. On this platform, *requesters* post tasks they want to be completed as *HITs* (Human Intelligence Tasks), to be performed by *workers*, also known as *turkers*. MTurk is the largest *crowdsourcing platform*, and has grown significantly over the past few years. The reason this work focuses on MTurk is that it is an ideal candidate for research into human computation - we identify three main advantages of studying this platform below.

The first advantage of using MTurk is the size of the platform. In the period between January 2009 and April 2010, Ipeirotis [30] tracked a total of slightly over 6.7 million HITs (Human Intelligence Tasks), with a total reward value of \$529,259. A more recent study by Difallah et al [11] using the same tool as the earlier work by Ipeirotis, MTurk Tracker, to track the MTurk market from 2009 to 2014, measured an average of 300K HITs available on the market at any given time, with an average of 10K new HITs arriving every hour. In addition, the available reward for work as well as the amount of active requesters at any time had also increased steadily since 2009. Such a large platform means there are large amounts of data to be collected for many different types of research.

[2]`http://www.crowdflower.com/`
[3]`https://www.duolingo.com/`
[4]`https://www.mturk.com/`

The second advantage of MTurk is the diversity of the platform, both in terms of the work and the workers. Using a classification of crowd work into six different task types by Gadiraju et al. [20], Difallah et al. showed that from 2009 until 2014, all of these types were represented, though naturally, not all were equally popular. There have been several publications researching the demographics of workers on MTurk, such as those by Ipeirotis [31] and Ross et al. [55], which show that on average, turkers are made up almost equally of men and women, turkers are of varying ages, and turkers are diverse in socio-economic factors like education, income and marital status as well as geographical origin, though a majority of workers are either from the United States or India.

The third main advantage of MTurk is that it has already been widely studied by researchers from many disciplines. Apart from being a popular research object for human computation researchers, MTurk has both been used as a tool and an object of research in fields such as human-computer interaction [38], social sciences [5], and many more. This means there is a wide body of work based on MTurk that can be used to inspire as well as support new research.

On MTurk, **requesters** offer work to be completed by **workers** in the form of **HITs** (Human Intelligence Tasks). We will consider each of these three dimensions of the platform in turn.

2.2.1 MTurk Requesters

To construct HITs, requesters can use templates provided by the MTurk platform on the website or through an assortment of tools provided by Amazon[5]. Alternatively, requesters can host external content as part of their HIT, allowing them to outsource HITs to websites like SurveyMonkey[6] or host their HIT on a web page of their own. This means that requesters using MTurk can ask workers to perform virtually any task that can be completed on a computer. Additionally, using MTurk is cheap, the worker population is very diverse and using MTurk to obtain data is not necessarily less reliable than using other, traditional methods [5] [49] [9].

The above does not mean that using MTurk is without its challenges, of course. These include the following:

- **Administration:** MTurk offers requesters a minimal crowdsourcing infrastructure, but the details of implementation are (understandably) left up to the requester. Due to this level of freedom, the requester has to consider many complicated issues, such as how much to pay workers, when to decide not to pay workers, or how to design a task interface, should the basic templates offered by MTurk not be adequate for the task the requester wishes to crowdsource. While there are requester communities online offering some idea on how to organize these things, there are few ‘golden rules’ when it comes to issues like wage setting or task design, leaving requesters having to find out a lot for themselves - a costly and time-consuming process.

[5]`https://requester.mturk.com/developer`
[6]`https://www.surveymonkey.com/`

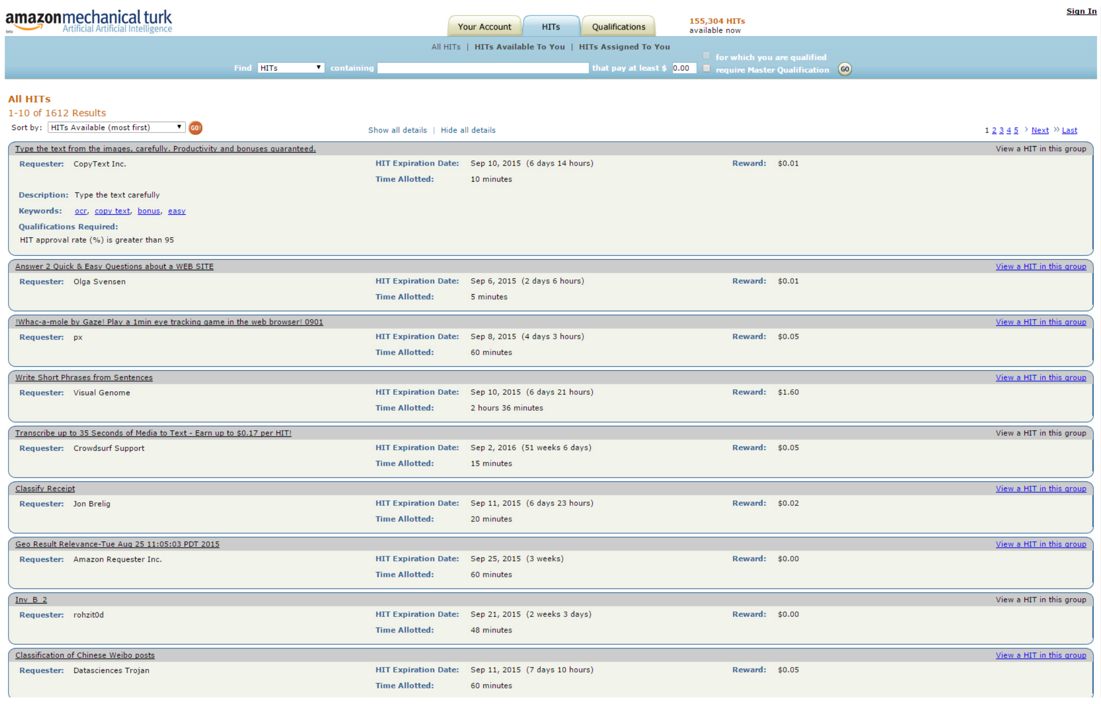
- **Worker Quality:** Because verifying the answers provided by workers is difficult, some workers try to take advantage of this by knowingly providing incorrect or incomplete answers to earn money more quickly. These workers are known as *spammers*. The challenge of *quality control* in crowdsourcing systems - trying to weed out spammers and low-quality workers while trying to obtain high-quality results at an acceptable cost - has been a popular topic of research over the past years, as described by, among others, Ipeirotis et al. [33], Difallah et al. [12] and Allahbakhsh et al [1].

- **Worker Motivation & Retention:** At one end of the worker spectrum, there are low-quality workers and spammers. At the other end of the spectrum, there are workers who consistently provide high-quality results. Ideally, requesters would like their worker population to consist of as many workers that are known to perform well as possible. This means that workers have to be attracted to a task first, motivated to do as much work as possible, and again motivated to come back at a later time when the requester has more work available. In the next subsection, we will discuss workers' perspectives on performing crowd work.

2.2.2 MTurk Workers

Workers can select which HITs to take on from a publicly available list on the MTurk homepage, though some of these HITs carry certain limitations: they might be restricted to workers that have a certain percentage of approved past work or workers that live in a certain country. A screenshot of the worker interface for finding HITs to work on is shown in Figure 2.1. In addition to this, some HITs are only intended for workers with certain qualifications that can be given out either by individual requesters or by Amazon. Overall, however, a typical worker who has a good record of their work being approved has a wide array of tasks available for them to complete at any time, meaning requesters have to make an effort to attract as many reliable workers as possible to their tasks.

Kauffmann, Schulze and Veit [37] provide an overview of worker motivation, dividing it into extrinsic and intrinsic motivation. Extrinsic motivation can be further divided into immediate payoffs ("earn money"), delayed payoffs ("improve skills"), and social motivation (e.g. a friend asks for your help), and intrinsic motivation can be divided into enjoyment-based ("creative outlet/fun") and community-based ("build a network of friends") motivation. While they found that the motivation for an immediate payoff was the driving factor, it was certainly not the only relevant one. This sentiment was echoed by, among others, Chandler and Kapelner, who noted a task's perceived meaningfulness seemed to have an effect on motivation to participate as well as the quantity and quality of output [10]. Rogstadius et al. [53] examined various types of extrinsic and intrinsic motivation, noting that increasing extrinsic reward might not always have a wholly positive effect on worker motivation: it might attract more workers, but will not significantly increase the quality of the work.


amazonmechanical turk
Artificial Artificial Intelligence
Your Account
HITs
Qualifications
Sign In
All HITs | HITs Available To You | HITs Assigned To You
Find
containing
that pay at least $
GO
Show all details | Hide all details
1 2 3 4 5 › Next » Last

155,304 HITs available now

Find HITs containing that pay at least $ 0.00
for which you are qualified
require Master Qualification

All HITs
1-10 of 1612 Results
Sort by: HITs Available (most first)

Type the text from the images, carefully. Productivity and bonuses guaranteed. — View a HIT in this group
Requester: CopyText Inc. — HIT Expiration Date: Sep 10, 2015 (6 days 14 hours) — Reward: $0.01
Time Allotted: 10 minutes
Description: Type the text carefully
Keywords: ocr, copy text, bonus, easy
Qualifications Required:
HIT approval rate (%) is greater than 95

Answer 2 Quick & Easy Questions about a WEB SITE — View a HIT in this group
Requester: Olga Svensen — HIT Expiration Date: Sep 6, 2015 (2 days 6 hours) — Reward: $0.01
Time Allotted: 5 minutes

!Whac-a-mole by Gaze! Play a 1min eye tracking game in the web browser! 0901 — View a HIT in this group
Requester: px — HIT Expiration Date: Sep 8, 2015 (4 days 3 hours) — Reward: $0.05
Time Allotted: 60 minutes

Write Short Phrases from Sentences — View a HIT in this group
Requester: Visual Genome — HIT Expiration Date: Sep 10, 2015 (6 days 21 hours) — Reward: $1.60
Time Allotted: 2 hours 36 minutes

Transcribe up to 35 Seconds of Media to Text - Earn up to $0.17 per HIT! — View a HIT in this group
Requester: Crowdsurf Support — HIT Expiration Date: Sep 2, 2016 (51 weeks 6 days) — Reward: $0.05
Time Allotted: 15 minutes

Classify Receipt — View a HIT in this group
Requester: Jon Brelig — HIT Expiration Date: Sep 11, 2015 (6 days 23 hours) — Reward: $0.02
Time Allotted: 20 minutes

Geo Result Relevance-Tue Aug 25 11:05:03 PDT 2015 — View a HIT in this group
Requester: Amazon Requester Inc. — HIT Expiration Date: Sep 25, 2015 (3 weeks) — Reward: $0.00
Time Allotted: 60 minutes

Inv_B_2 — View a HIT in this group
Requester: rohzit0d — HIT Expiration Date: Sep 21, 2015 (2 weeks 3 days) — Reward: $0.00
Time Allotted: 48 minutes

Classification of Chinese Weibo posts — View a HIT in this group
Requester: Datasciences Trojan — HIT Expiration Date: Sep 11, 2015 (7 days 10 hours) — Reward: $0.05
Time Allotted: 60 minutes

Figure 2.1: AMT Worker Interface.

Apart from motivation, workers also have other concerns, several of which are outlined by Irani et al [34]. For example, some workers feel that their work is sometimes rejected unfairly or arbitrarily, that they are sometimes not paid enough or on time, and that employers and Amazon do not respond to their concerns. This encourages Irani et al to introduce Turkopticon[7], a tool that workers can use to score requesters' reliability. The complaints registered by Irani et al. can also be found when browsing online worker communities like Turker Nation[8], MTurk Forum[9] and certain subreddits[10]. Besides low pay, unfair treatment and unclear communication by requesters consistently come forward as the most important issues affecting workers.

2.2.3 MTurk HITs

Taking into account the motivations of and challenges faced by both requesters and workers, we turn our attention to the medium through which they interact: the HIT.

Requesters want to design HITs in a way that attracts high-quality workers and makes it easy to detect low-quality answers and spammers. Workers want to work on clearly defined tasks with unambiguous criteria for evaluation and payment, and feel like they are being treated fairly. Several tools exist to help requesters build better HITs: these include psiTurk[11], which is focused on behavioural research, and Turkit [43]. There are also many blogs online that offer requesters advice. However,

[7]`https://turkopticon.ucsd.edu/`
[8]`http://turkernation.com/`
[9]`http://www.mturkforum.com/`
[10]`http://www.reddit.com/r/mturk`
[11]`https://psiturk.org/`

current best practices in MTurk task design seem to be based mostly on personal experience and general consensus, rather than concrete evidence of efficacy.

We believe that proper task design can have a massive impact on the crowdsourcing process. In the next section, we will elaborate on several aspects of task design that have already been subject to research, and identify a gap in the research so far linking back to our original research questions.

2.3 Aspects of Task Design

In this section, we will summarize briefly previous work into different aspects of task design. First, we will describe several previously researched task features, after which we summarize the works we investigated. Then, we will identify research opportunities based on this previous work.

2.3.1 Common Features

Since there are many different variables involved in task design that might influence the outcome of an experiment, a common approach is to keep a large amount of these variables constant and compare situations in which only a single or a few variables are different. For the purposes of categorizing existing work, we will consider the following dimensions:

- **Single-Session or Multi-Session** Whether or not the experiment is carried out in a single session (workers participate once) or multiple sessions (workers participate, then return at a later time to do more work, measuring worker retention).
- **Task Type Variation** - Whether or not the type of task workers performed was varied in the experiments. For example: workers are put in groups A and B: those in group A perform an annotation task, and workers in group B review work by other workers in addition to performing the annotation task themselves. We do not consider things like including demographical surveys or control questions to be variation in this sense.
- **Task Platform Variation** - Whether or not different workers performed their tasks in different contexts. We define a crowdsourcing platform as the medium for a crowdsourcing transaction such as a crowdsourcing platform like Mechanical Turk, a citizen science website or a mobile app.
- **Task Difficulty Variation** - Whether or not the cognitive difficulty of the task was varied among groups of workers. This does not include similar cases, such as one in which all workers perform the same task, which increases in difficulty as time passes.
- **Motivational Strategies** - Whether or not non-financial motivational strategies were used to engage workers. This includes strategies like gamification of the task or feedback on quality of work or rank.
- **Pay Amount Variation** - Whether or not the monetary reward for the experiment was varied among groups of workers.

- **Pay Structure Variation** - Whether or not the structure of pay was varied among groups of workers, for example paying some workers for time spent working while paying other workers for every task they complete.
- **Other Incentives** - Whether or not non-financial incentives were used as a form of payment. Examples could include access to a certain service or virtual items in a video game. This does not include experiments in which workers were volunteers and thus did not directly receive any tangible reward.

Table 2.1 shows an overview of the work discussed in the next subsection, categorized according to these features.

2.3.2 Previous Work

Table 2.1: Overview of Task Design-Related Research

Authors	**Year**	**Single or Multi-Session**	**Task Type**	**Task Plat-form**	**Task Diffi-culty**	**Motivational Strategies**	**Pay Amount**	**Pay Struc-ture**	**Other Incen-tives**
Mason, Watts	2010	S			x		x	x	
Barankay	2011	M				x			
Rogstadius et al.	2011	S				x	x		
Shaw et al.	2011	S				x	x	x	
Eickhoff et al.	2012	S	x				x		(x)
Dow et al.	2012	S	x			x			
Lee et al.	2013	M				x			
Goncalves et al.	2013	S		x	x	x	x		
Huang, Fu	2013	S				x			
Chandler, Kapelner	2013	S				x			
Mao et al.	2013	S					x	x	
Yu et al.	2014	M	x			x	x	x	
Tomasic et al.	2014	M				x			x
Ipeirotis, Gabrilovich	2014	S		x		x	x	x	

Mason and Watts [45] tested the effect of financial incentives on worker performance. Their first experiment asked workers to sort pictures of moving traffic in temporal order, and there were 3 levels each of pay rate and task difficulty. Their second experiment involved workers solving a word puzzle and being paid either per word or per entire puzzle. Task difficulty and pay structure and amount were thus varied, while task type, context and motivational strategies were constant.

Barankay [2] explored the effects of telling workers how their performance ranks compared to others. First, he posted two tasks on MTurk, with one indicating workers who perform it will receive feedback. Second, he invited the workers from the first stage to perform more work, while randomly including feedback on the first task in the invitation sent to half of the participants. Finally, he asked workers to complete a survey. Barankay only varied motivational strategies - all of our other variables were constant throughout his experiments.

Rogstadius et al. [53] performed a 2x3 experiment where workers were paid different amounts and told they were working for a non-profit or for-profit requester. The task involved identifying malaria cells in an image. Task type, context, difficulty and context as well as pay structure were constant, while motivational strategies and pay amount were not.

Shaw et al. [56] tested a wide variety of social and financial strategies to improve worker performance. The task consisted of answering several qualitative questions about the design of a website. The researchers used different social motivational strategies and varied pay structure and amount.

Eickhoff et al. [16] also compared the gamified version of a task to its standard counterpart. Workers were asked to vote on the relevance between two topics, with some workers performing it as a traditional HIT and others playing an annotation game. The type of task was thus different in these two conditions, while difficulty and context were the same. Pay structure was constant, but the amount was not: workers in the gamified task were paid less, with the authors specifically noting the fun from playing the game to be a part of the reward the workers received.

Dow et al. [14] investigated whether personalized feedback helps workers persevere and produce better results. Workers were asked to write reviews for one product each in 6 of 20 categories. Then, they received no feedback, and were then only able to edit their review after they had filled out a self-reflection, or had received feedback from an external expert before being given the option to edit their reviews. The researchers varied motivational strategies (different types of feedback) and task type (self-evaluation), while task difficulty and context as well as pay amount and structure were constant.

Lee et al. [42] investigated the effect of different types of performance feedback and gamification on crowdsourced worker engagement. The context for the experiment was a platform at IBM where users are asked to identify if Twitter accounts belong to a certain IBM employee. In a 2x3 experiment, they showed workers their own achievements, the contributions by the community, or a combination of the two, and each of these modes was carried out in two ways: gamified and non-gamified. This way, the researchers varied different motivational strategies while keeping all of the other variables we are interested in constant.

Goncalves et al. [21] placed a public display at a university, offering passers-by a task that was also performed by workers on MTurk. The workers performed the same malaria image annotation task used by Rogstadius et al. This work varied pay amount (volunteers versus different rates for turkers), motivational strategies (different types of motivating text), task difficulty and task context (public display versus crowdsourcing platform), while keeping pay structure and the type of task constant.

Huang and Fu [29] investigated the effects of social transparency on worker performance. Workers were asked to count the amount of nouns in a list of 30 words, and worked either individually, competitively against another worker, or as a team with another worker. In the final two cases, workers were either anonymous or their demographic information was disclosed to the other worker. Keeping every other aspect constant, Huang and Fu tested different motivational strategies based on teamwork and social transparency.

Chandler and Kapelner [10] tested motivational factors of workers on Mechanical Turk by constructing several testing scenarios with differing levels of expected motivational value. The task was to identify objects of interest in an image: some workers were told only this, while others were told they were labeling cancerous cells to assist medical researchers. With this testing set-up, Chandler and Kapelner only varied motivational strategies.

Mao et al. [44] compared the performance of volunteers using the Galaxy Zoo citizen science project to the performance of turkers rewarded according to one of three different, non-performance contingent payment schemes. Varying the payment schemes also included a difference in pay amount (different schemes lead to different average hourly wages). Task type, difficulty, context (since the volunteer data was a baseline, not part of the experiment) and motivational strategies were not varied.

Yu et al. [64] wanted to improve worker engagement and output quality. They measured engagement by the *return rate*, defined as the percentage of workers that, when invited, returned to perform more work for the researchers after having worked for them before. The task consisted of summarizing portions of scientific papers: quality was assessed by experts at first, and also by crowd workers in later experiments. The researchers tried out three types of motivational strategies: social, learning and financial. In addition, they tested the effects of these strategies not only separately but also investigated the effects of using more than one strategy at once. Task difficulty and context were constant, but across their three experiments they varied pay amount and structure (financial strategies), motivational strategies (social and learning), and the type of task (part of the learning strategy involved giving feedback to others).

Tomasic et al. [57] wanted to motivate workers to contribute data to a participatory sensing system, the system being an application that tracked arrival information for public transit services based on location traces supplied by users. The workers were the users of the application, and they were not paid in money: instead, some workers were requested to supply trace data, and other workers were denied access to the app's functionalities if they did not contribute enough data. This experiment varied motivational strategies and provided non-financial incentives, while task type, difficulty and context as well as payment were constant.

Finally, Ipeirotis and Gabrilovich [32] tried to compare the performance and engagement of paid and volunteer workers. Their paid workers were recruited from Mechanical Turk or oDesk, while the volunteers were recruited through online advertising. The task involved answering trivia questions about various subjects. Task difficulty and the type of task were constant, and pay amount (volunteers versus paid), pay structure (difference between MTurk and oDesk), motivational strategies (various types of feedback) and task context (crowdsourcing platform versus quiz website) were varied.

Considering the existing work, it seems most research into task design employs motivational strategies or variations in pay with the goal to improve worker motivation and result quality. There is a little investigation into *task difficulty*, but we feel that previous research into task difficulty misses a crucial attribute of a task: its presentation. In other words, the actual code making up the web page that the task takes place on. We identify a gap in the literature concerning the full *complexity* of human intelligence tasks: in addition to aspects of task design involving reward structures and motivational strategies, research into HIT design should incorporate a wider variety of measures for complexity. In the next section, we give our definition of complexity and the various dimensions we will explore in this thesis.

2.4 Measuring Task Complexity

2.4.1 Introduction

For several weeks, we observed how many tasks posted on MTurk were externally hosted, and thus were custom-made. We identified these tasks by the presence of an ExternalQuestion iframe on the HIT page, and observed daily percentages between sixty-five and eighty percent of externally hosted HITs. Based on our measurements, we estimate that on average over two thirds of HITs available at a given moment are externally hosted. We believe this is mainly due to the presence of a small amount of requesters who each post a large amount of tasks for which they have developed a custom interface. These HITs typically involve tasks that are not suited to the standard template MTurk offers, such as audio or video transcription.

Since such a large amount of HITs involves a custom-made interface, we believe it might be worthwhile to investigate how the structure and cognitive complexity of such HITs relates to the challenges outlined in the previous sections, such as worker motivation and retention, gold standards for task design, spammer detection and more. This might lead to insights about the complexity of HITs. Campbell [7] presents a framework for objectively looking at task complexity, stating that *"Any objective task characteristic that implies an increase in information load, information diversity, or rate of information change can be considered a contributor to complexity."* We will apply this proposed framework to the case of HITs on MTurk, and define the concept behind measuring the complexity of a HIT for the purposes of our research as follows:

*The **complexity** of a HIT can be approximated by the sum of the complexity of its objectively measurable attributes, as perceived by its workers.*

HITs certainly also have subjective characteristics. For example, a worker might find a task easier if they have prior experience with a similar task. While this is certainly worth investigating, this thesis will focus on the objectively measurable characteristics of tasks. In this section, we will first look at the background of task complexity based on Campbell's survey [7]. Then, we will look at related work that investigated web page complexity specifically.

2.4.2 Task Complexity

Campbell [7] describes that in the various fields that investigate task complexity, *"Complexity is treated as: (a) primarily a psychological experience, (b) an interaction between task and person characteristics, and (c) a function of objective task characteristics."* Researchers typically approach task complexity from one of these three viewpoints, and the task characteristics used to measure complexity differ for each viewpoint. Viewing complexity as a psychological experience focuses on the perception of the worker of the task: characteristics like the challenge, arousal and stimulation generated by the task are considered. When complexity is viewed as a task-person interaction, the mental state and history of the person are important: characteristics like the experience, interest and attention span of a worker play an important role in addition to the difficulty of the task itself. Finally, viewing complexity as a function of objective task characteristics implies metrics that relate more directly to the way the task is presented: how high is the information load generated? Are there multiple paths to arrive at the desired end-state, and how clearly are they presented?

We consider the third viewpoint - task complexity as a function of objective task characteristics - to be the most relevant to this thesis. An important notion, forwarded by Campbell and Gingrich [8] among others, is that *complex tasks place a high cognitive demand on the individual*: a task can have inherent characteristics that make it more complex which do not depend on the individual completing the task. For this reason, we are interested in relating HIT complexity in some way to concepts like cognitive load and information load.

2.4.3 Web Page Complexity

One of the most important contributors to the perceived complexity of a task is the way it is presented - in the case of HITs, as a web page. Therefore, existing ways to analyze the complexity of web pages could prove useful in trying to define the complexity of a HIT.

Several attempts have already been made to attempt to categorize web page complexity. In 2001, Ivory et al. [36] analyzed a large amount of websites and defined several page-level metrics to predict whether human judges would rate the design of the websites as good or bad. They summarize their metrics as follows: the metrics *"concern page composition (e.g., word count, link count, graphic count), page formatting (e.g., emphasized text, text positioning, and text clusters), and overall page characteristics (e.g., page size and download speed)."* The metrics were chosen based on an earlier study by the same researchers in which they were linked with effective design and usability [35].

In 2007, Nadkarni and Gupta [48] proposed that *perceived website complexity* (PWC) is central to understanding how the way information is presented to website users affects their experience. Based on a review of web design literature, they defined thirteen objective complexity measures in addition to a large amount of subjective metrics based on the user experience, measured through surveys. The objective measures included the visual characteristics of a page (percentage of white space, amount of graphics, color count), the information presented (word count, different types of presentation forms), and the interrelations a page shows with other resources (links to other pages, amount other webpages configuring a certain page).

In 2011, Butkiewicz et al. [6] identified a set of metrics to characterize website complexity. Some metrics were based on content (number and types of objects, bytes downloaded), while others were based on what the researchers term *service complexity*. These metrics included the amount of servers involved in presenting a website and the amount of content fetched from somewhere other than the origin of the page.

Apart from these efforts to define web page complexity, there has also been some focus on the visual complexity of web pages. Harper et al. [23] specifically note it as being an implicit measure of cognitive load, their reasoning being that *"by understanding a user's visual perception of Web page complexity, we can understand the cognitive effort required for interaction with that page."* They performed a study asking human judges to compare web pages, and eventually sorted the pages into three categories: simple, neutral and complex web pages. Simple pages tended to contain information about a single subject, and feature small amounts of items like tables, boxes and lists in their layout. As pages grew more complex, they tended to be longer and have larger numbers of images, links, tables and menus. In short: simple pages feature less diverse, shorter content and simple colors, while complex pages are long, have a lot of design elements, and have a high intensity and variety of fonts and colors.

In 2013, Harper et al. [22] constructed an algorithm that identifies areas of a web page that are highly complex by dividing a page into 'chunks' of block-level elements. The algorithm showed a correlation of 86% with user judgements of complexity. Their results supported their findings from their previous work, in that they seemed to indicate once again that more structural elements imply a more (visually) complex web page.

Finally, also in 2013, Reinecke et al. [52] constructed models to measure perceived visual complexity and colorfulness of websites. They argue that visual complexity is an important factor in users' first impressions of a website, which they evaluated by comparing the predictions by their computational models with ratings given by online human judges. The main metrics used fell into the categories of color (closeness to W3C color, average pixel value in the HSV color space), space-based decomposition (amount of text groups, text area and non-text area) and quadtree decomposition (subdividing an image into blocks that are more homogeneous than the image itself, then analyzing those blocks).

Apart from the different metrics and approaches used, the message is the same: many objectively measurable characteristics of a task or web page can influence complexity. We believe we can draw on this existing work to investigate the complexity of HITs, which are tasks that take place on web pages. In the next chapters, we will describe the approach we used to analyse HIT complexity and the results of our research.

Chapter 3

Automation of Task Complexity Analysis

In this chapter, we will describe our approach towards measuring task attributes and task complexity. First, we describe the task attributes we selected for evaluation and our reasoning for selecting them. Second, we describe the process by which we obtained measurements for each of these attributes from the data. Finally, we describe the design of the human intelligence task we used to obtain complexity judgements from human workers.

3.1 Attribute Selection

3.1.1 Introduction

Using data gathered from tracking the MTurk HIT market, we can select a number of task attributes that we consider to have potential in proving a link between task attributes and task complexity. These attributes can include ones that are measured by the tracker and can be used in the evaluation without further effort, and they can also include attributes extracted from market data after it is stored, or more specifically from the HTML content of HITs. In the following subsection, we outline the set of attributes we chose to evaluate based on logic, experience and literature. We considered which task attributes to evaluate based on the following requirements:

1. The attribute should be **unambiguous** and **objectively measurable**. Based on its description, it should be clear how an attribute could be extracted from MTurk market data.
2. The attribute should be **related to task complexity** in some way. As defined by Campbell, increased complexity implies *"an increase in information load, information diversity, or rate of information"*. This relationship should be present for every attribute we select, based on existing literature and logic.

In general, we focused on quantitative attributes over qualitative ones. The attributes we consider tend to be more concrete than abstract and are straightforward to measure. However, we also include several semantic features that are slightly higher-level than the others, such as term frequency of unigrams in HIT content and text topic features. We also considered including more abstract attributes such as page navigability or task difficulty, but due to the time required to correctly implement measurement of such attributes we opted to focus on the set of attributes presented in this section. In Chapter 6, we will reflect on our selection and identify possibilities for future research in task attribute.

The attributes we consider are divided into three categories. The first category, **Metadata**, consists of attributes that are unrelated to the content of a HIT group, but are instead made up of task metadata like the required qualifications or the HIT description. These are the attributes that are shown in the MTurk interface to workers before they start work on a HIT. We believe that attributes like the reward or allotted time to complete a task might be able to provide an indication of the level of complexity of that task. In this category, we consider attributes that were also used by Difallah et al [11].

The second category, **Content Attributes**, contains attributes that relate to the HTML content of the HIT group, or in other words, the actual body of the task. We believe that attributes like the amount of external content used on a HTML page might serve as indicators for page complexity, and that we can use traditional measures of website complexity in measuring HIT complexity, as HITs are presented to and interacted with by the user in the same way as a commercial website. In this category, we consider website complexity measures suggested by Ivory et al [36] and Butkiewicz et al [6]. Additionally, work in the domain of community question-answering systems, by among others Ravi et al. [51] and Yang et al. [62], suggests that task topic could be indicative of the quality of task specification. How well a task is formulated might influence workers' perception of a task, and therefore could be indicative of its complexity. For these reasons, we also consider a small set of semantic features, focusing on the use of keywords, words in task content, and the topic(s) of the text.

The third category, **Visual Attributes**, focuses on attributes that relate to the visual complexity of a page. We believe that measures like the amount of visual areas or the colorfulness of a page might help indicate when a HIT is complex because of its visual layout. In this category, we consider visual complexity measures used by Reinecke et al [52].

For each selected attribute, we give its name, describe it, explain our rationale for selecting this attribute, and formulate a hypothesis about the nature of the attribute in relation to our research. Table 3.1 provides an overview of the attributes and for each one lists a short description and the related work which inspired us to incorporate it in our own research. Metadata attributes were collected directly from the MTurk interface, Content attributes were extracted from web pages, and Visual attributes were extracted by using code provided by Reinecke et al. Examples of a quadtree-based and space-based decomposition of a single HIT are shown in Figure 3.1. These decompositions were used in measuring attributes A22, A23 and A24.

Attribute	**Description**	**Related Work**
A1: Reward	The reward for completing the task in cents.	Difallah, Faridani
A2: Time Allotted	The time allotted to a worker in which they should complete a task, in seconds.	Difallah
A3: Location	Whether or not workers are required to have a certain geographical location.	Difallah
A4: Master	Whether or not workers are required to have a Master qualification.	Difallah
A5: Total Approved	The minimum amount of approved tasks workers are required to have.	Difallah
A6: Approval Rate	The minimal percentage of approved tasks workers are required to have.	Difallah
A7: Title Length	Length of the HIT title.	Difallah
A8: Description Length	Length of the description string in the HIT preview box.	Difallah, Campbell
A9: Amount of Keywords	Amount of keyword strings in the HIT preview box.	Difallah
A10: HITs Available	Amount of HITs available for this HIT group	Difallah
A11: Link Count	The amount of links in the body text.	Ivory
A12: Word Count	The amount of words in the body text.	Ivory
A13: Image Count	The amount of images on the page.	Ivory, Butkiewicz
A14: Body Text Percentage	Percentage of words that are body rather than display text (i.e., headers).	Ivory
A15: Emphasized Body Text Percentage	Percentage of words that are emphasized rather than plain text.	Ivory
A16: Amount of Stylesheet Files	The amount of stylesheet files included in the page.	Butkiewicz
A17: Amount of Script Files	The amount of script files included in <script> tags on the page.	Butkiewicz
A18: Amount of Input Elements	The amount of input elements (e.g. text fields, buttons..) on the page.	Campbell
A19: Keywords	Binary features of keywords provided by the requester.	Yang, Ravi
A20: Unigrams	TF-IDF features of single words used in task title and content.	Yang, Ravi
A21: Topics	LDA topic model features of the unigrams found in task title and content.	Yang, Ravi
A22: Amount of Text Groups	Amount of horizontal groups of text characters. Each group may represent a word, a sentence, or a paragraph.	Reinecke
A23: Amount of Image Areas	Amount of image areas. Several adjacent images are counted as one image area.	Reinecke
A24: Amount of Visual Areas (leaves in quadtree)	The amount of visual areas obtained by recursively dividing a task presentation into regions until a region has no visual space.	Reinecke, Zheng
A25: W3C Colors	Percentage of pixels that are close to one of the 16 colors defined by the W3C.	Reinecke, Rosenholtz
A26-H: Hue, A26-S: Saturation, A26-V: Value	The average HSV value of pixels.	Reinecke
A27: Yendrikhovskij Colorfulness	Colorfulness as defined by Yendrikhovskij.	Yendrikhovskij, Reinecke
A28: Hasler Colorfulness	Colorfulness as defined by Hasler.	Hasler, Reinecke

Table 3.1: Overview of Task Attributes

3.1.2 Description of Measured Task Attributes

Metadata Attributes

Attribute A1: Reward

Description: The reward for completing the task in cents.

Rationale: Faradani et al [18] note that requesters commonly set the rewards for their HITs based on rules of thumb. One of the most prominent of these rules of thumb is selecting an hourly wage that is considered fair, estimating the required time to complete a task, and setting the reward based on that. This rule can be found on many blog posts online [1]. Faradani et al also note a trade-off between reward and completion time. Additionally, Difallah et al. used the reward attribute in their predictions of throughput. We believe that more complex tasks can often be expected to take longer to complete than tasks with a lower complexity, and if this is the case, their rewards should also be greater.

Hypothesis H1: *The complexity of a task affects the size of its reward.*

Attribute A2: Time Allotted

Description: The time allotted to a worker in which they should complete a task, in seconds.

Rationale: This attribute was already used by Difallah et al to predict throughput. Building on the rationale for attribute A1, a requester that expects a task to take longer will set a higher time allotted value. Therefore, we hypothesize that more complex tasks will have more time allotted to complete them.

Hypothesis H2: *The complexity of a task affects its allotted completion time.*

Attribute A3: Location

Description: Whether or not workers are required to have a certain geographical location.

Rationale: Location requirements are typically used to restrict a HIT to US-based workers. US-based workers are expected to perform better on tasks, even those that do not rely mainly on command of the English language. This attribute was also used by Difallah et al.

Hypothesis H3: *The complexity of a task affects whether or not the task requires a certain worker location.*

[1] `http://neerajkumar.org/writings/mturk/`, Question 3

Attribute A4: Master

Description: Whether or not workers are required to have a Master qualification.

Rationale: Master workers are given this qualification by Amazon when they consistently perform well in certain task categories. Requesters can then restrict their tasks to master workers (at a premium), from whom they can expect higher quality results. This is possibly because they feel the task would be too complex for non-master workers. This attribute was also considered by Difallah et al.

Hypothesis H4: *The complexity of a task affects whether or not the task requires a certain worker location.*

Attribute A5: Total Approved

Description: The minimum amount of approved tasks workers are required to have.

Rationale: Similar to attribute A4 and used by Difallah et al., requesters expect workers with a higher amount of approved tasks to have more experience and thus perform better on HITs.

Hypothesis H5: *The complexity of a task affects the minimum amount of approved tasks workers are required to have.*

Attribute A6: Approval Rate

Description: The minimal percentage of approved tasks workers are required to have.

Rationale: Similar to A4 and A5 and also considered by Difallah et al., if a worker has a high percentage of approved tasks, they are expected to deliver high quality results.

Hypothesis H6: *The complexity of a task affects the minimum percentage of approved tasks workers are required to have.*

Attribute A7: Title Length

Description: The length of the title string shown in the HIT preview box.

Rationale: Also used by Difallah et al, the title is a short summary of the task. A more complex task might be more difficult for requesters to summarize, which would lead to longer titles for more complex tasks.

Hypothesis H7: *The complexity of a task affects the length of its title.*

Attribute A8: Description Length

Description: The length of the description string in the HIT preview box.

Rationale: Considering Campbell's definition, description length is one of the first task attributes the worker sees that could contribute to complexity. An overly long description might be indicative of a task that is perceived as complex, and also in itself contribute to the perception of a task as complex.

Hypothesis H8: *The complexity of a task affects its description length.*

Attribute A9: Amount of Keywords

Description: The amount of keyword strings in the HIT preview box.

Rationale: Similar to the reasoning behind including attributes A7 and A8, an increased amount of keywords might imply a larger cognitive load associated with a task: as a task gets more complex, requesters would need to include more keywords to summarize its content.

Hypothesis H9: *The complexity of a task affects the amount of keywords in its preview.*

Attribute A10: HITs Available

Description: Amount of HITs available for this HIT group.

Rationale: When Difallah et al. attempted to predict HIT batch throughput, they identified the amount of available HITs as having a significant impact on the prediction score. We include it to see if our data supports their findings. If it is the case that tasks with larger batches are more attractive to workers, requesters could consider offering larger batches of complex tasks or mixing tasks of various complexity into a single batch to achieve a faster average completion time.

Hypothesis H10: *The amount of HITs available in a batch correlates with the throughput of that batch.*

Content Attributes

Attribute A11: Link Count

Description: The amount of links in the body text.

Rationale: This metric was proposed by Ivory et al in their study on empirical web page design metrics. A higher amount of links to external pages implies that the information the worker needs might not all be on the current page, meaning they will have to navigate to other pages and put in more effort to complete their task. Even if the links are not directly relevant to the task (such as a 'contact us' button), they increase the cognitive load by adding yet another element to the page.

Hypothesis H11: *The complexity of a task affects the amount of links used in its content.*

Attribute A12: Word Count

Description: The word count of the text of the page.

Rationale: This metric was proposed by Ivory et al in their study on empirical web page design metrics. A higher word count implies a greater information load, meaning the worker has to put in more effort to understand and complete their task.

Hypothesis H12: *The complexity of a task affects the word count in its content.*

Attribute A13: Amount of Images

Description: The amount of images on the page.

Rationale: Following Campbell's framework, including more images on a page should contribute to the cognitive complexity experienced by the user. This metric was also proposed by Ivory et al in their study on empirical web page design metrics, and Butkiewicz et al. argue the amount of objects on a page and their MIME types can contribute to complexity. We expect that images included in HIT content will in nearly all cases be relevant to the task at hand. In addition, reliably distinguishing non-relevant from relevant images would most likely require going through our entire dataset manually. Thus, for both logical and practical reasons, we consider the amount of images as a whole and do not distinguish between different kinds of images in this case.

Hypothesis H13: *The complexity of a task affects the amount of images used in its content.*

Attribute A14: Body Text Percentage

Description: Percentage of words in text that are body rather than display text (i.e., headers).

Rationale: This metric was proposed by Ivory et al in their study on empirical web page design metrics. A higher percentage of body text indicates relatively less guidance on the page for the worker to make sense of the task and relatively more text on the page to read, possibly leading to higher complexity.

Hypothesis H14: *The complexity of a task affects the percentage of body text relative to display text in its content.*

Attribute A15: Emphasized Body Text Percentage

Description: Percentage of words in text that are emphasized.

Rationale: This metric was proposed by Ivory et al in their study on empirical web page design metrics. Similarly to A13, a higher emphasized body text percentage implies the page designer has put in effort to manage the user experience. Though a very large body text percentage is most likely indicative of bad design (such as an entire page of text being bolded), we expect that typically emphasized body text should have a slightly diminishing effect on complexity.

Hypothesis H15: *The complexity of a task affects the percentage of emphasized body text in its content.*

Attribute A16: Amount of Stylesheet Files

Description: The amount of stylesheet files included in the page.

Rationale: A stylesheet is used to improve the presentation of a website. On the one hand, when done properly this might reduce the cognitive complexity by streamlining the user experience. On the other hand, it might increase cognitive load, as argued by Butkiewicz et al.

Hypothesis H16: *The HTML content of tasks that are more complex will contain more stylesheet files.*

Attribute A17: Amount of Script Files

Description: The amount of script files included in <script> tags on the page.

Rationale: Scripts are used to provide enhanced functionality. In Butkiewicz et al's research, over half of the pages they investigated fetched more than six JavaScript files. Following their reasoning, increasing the amount of JavaScript linked to by a page might be an indicator for increased complexity.

Hypothesis H17: *The complexity of a task affects the amount of script files used in its content.*

Attribute A18: Amount of Input Elements.

Description: Amount of HTML form input elements on the page, including <input>, <textarea>, <select> and <button>.

Rationale: Following Campbell's framework, including more input elements on a page should contribute to the cognitive complexity experienced by the user.

Hypothesis H18: *The complexity of a task affects the amount of input elements used in its content.*

Attribute A19: Keywords

Description: Binary features of keywords provided by the requester.

Rationale: We might be able to see a difference in task complexity between tasks based on semantic features. One way of trying to find a difference in task types could be through task keywords, which in most cases should be indicative of a task's content. For example, we might be able to find that complex tasks tend to feature certain keywords more or less often.

Hypothesis H19: *The complexity of a task affects the keywords used in its description.*

Attribute A20: Unigrams

Description: TF-IDF features of single words used in task title and content

Rationale: Ravi et al. [51] used unigram features to predict the quality of a question. Similarly, we might be able to use unigrams to predict the complexity of HITs. We can use tf-idf (term frequency - inverse document frequency) to identify the 'most important' unigrams in specific HITs, using our entire set of HITs as a corpus on which to base the measurement. While specific unigram frequency may not be as indicative of the semantics of a task as a single keyword (because keywords are fewer in number and more specific), we consider all unigrams of a task as a whole. By doing this, we might find that specific terms are linked to task throughput.

Hypothesis H20: *The complexity of a task affects the frequency of unigrams used in its title and content.*

Attribute A21: Topics

Description: LDA topic model features of the unigrams found in task title and content.

Rationale: Ravi et al. not only used unigrams, but also LDA [4] topic models to predict question quality. In a sense, topics are a more compressed version of unigrams, because it extracts the most important concepts from the unigrams relative to the entire corpus.

Hypothesis H21: *The complexity of a task affects the topic weight of the topics found in its title and content.*

Visual Attributes

Attribute A22: Amount of Text Groups

Description: Amount of horizontal groups of text characters. Each group may represent a word, a sentence, or a paragraph.

Rationale: A larger amount of text groups implies that the visual complexity of a page will be higher, as users will have to divide their attention among the text groups.

Hypothesis H22: *The complexity of a task affects the amount of text groups in its user interface.*

Attribute A23: Amount of Image Areas

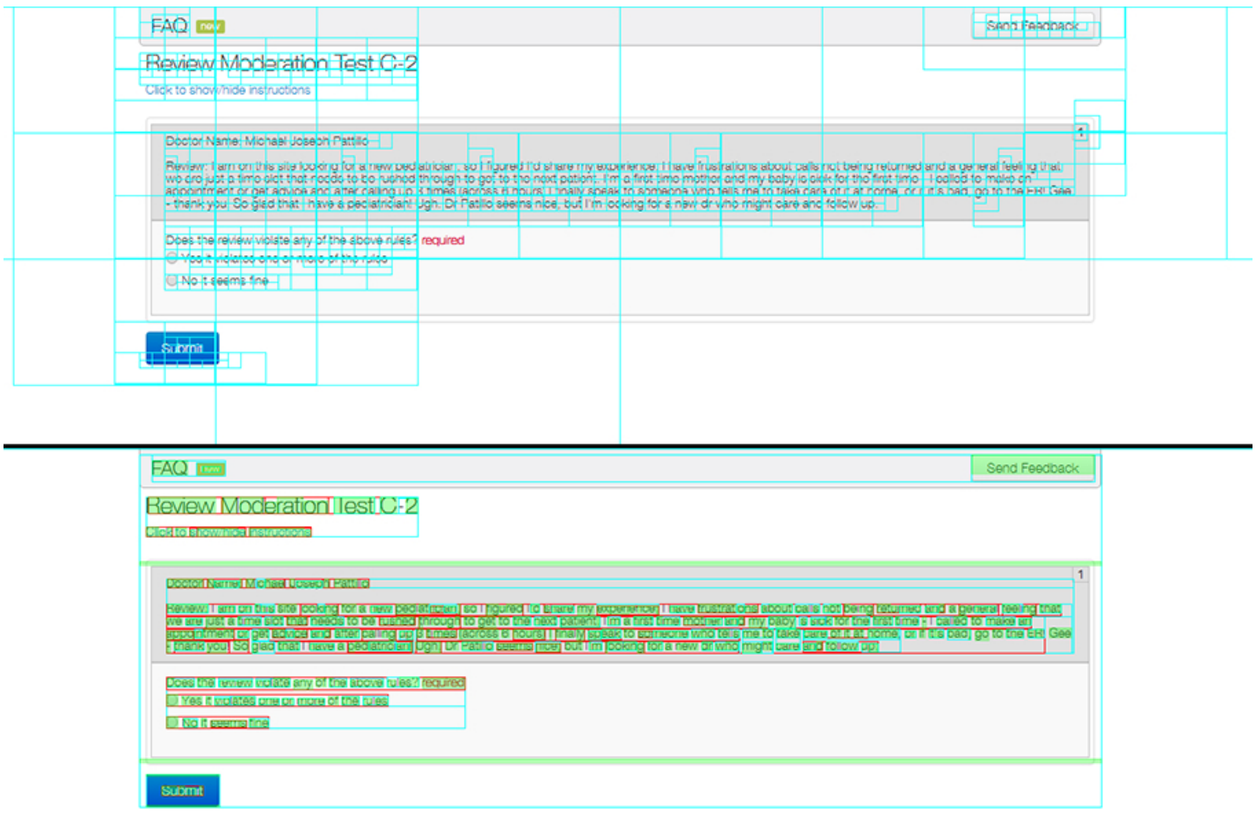
Description: Amount of image areas on the page. Several adjacent images are counted as one image area.

Rationale: Similar to the rationale for attribute A19, a higher amount of image areas should increase cognitive load on the user, and thus increase the task's visual complexity.

Hypothesis H23: *The complexity of a task affects the amount of image areas in its user interface.*

Attribute A24: Amount of Visual Areas (leaves in quadtree)

Description: The amount of visual areas obtained by recursively dividing a task presentation into regions until a region has no visual space.



Doctor Name: Michael Joseph Pattilo

Review: I am on this site looking for a new pediatrician, so I figured I'd share my experience. I have frustrations about calls not being returned and a general feeling that we are just a time slot that needs to be rushed through to get to the next patient. I'm a first time mother and my baby is sick for the first time - I called to make an appointment or get advice and after calling up 3 times (across 6 hours) I finally speak to someone who tells me to take care of it at home, or if it's bad, go to the ER! Gee - thank you. So glad that I have a pediatrician! Ugh. Dr Pattilo seems nice, but I'm looking for a new dr who might care and follow up.

Does the review violate any of the above rules? required

Yes it violates one or more of the rules

No it seems fine

Figure 3.1: Visual Decompositions of a HIT.
Above: quadtree-based decomposition. Below: space-based decomposition.

Rationale: Reinecke et al. refer to work by Zheng et al. [66] that indicated that the higher the amount of leaves in the quadtree (which we refer to as visual areas) was, the more likely websites were to be judged by users as complicated and unprofessional - or in other words, more visually complex. Reinecke et al.'s own observations did not confirm this connection to be as strong as in the work by Zheng et al., but they do state it partially contributes to visual complexity.

Hypothesis H24: *The complexity of a task affects the amount of visual areas in its user interface.*

Attribute A25: W3C Colors

Description: Percentage of pixels that are close to one of the 16 colors defined by the W3C.

Rationale: Reinecke et al consider various metrics of colorfulness, this being one of them. They refer to work by Rosenholtz et al. [54], who define visual *clutter* as being an important contributing factor to visual complexity, and consider that metrics such as a largely colorful interface or a high color variability can increase the perceived sense of clutter.

Hypothesis H25: *The complexity of a task affects the percentage of pixels in its user interface that is close to a W3C color.*

Attribute A26-H, A26-S, A26-V: Hue, Saturation, Value

Description: The average HSV value of pixels, each considered separately.

Rationale: As stated by Reinecke et al., color can cause a sense of clutter but also influence features such as the perceived trustworthiness of a page. For this reason, we consider that the average values of the different color dimensions of a page - hue, saturation and value - might be indicative of visual complexity.

Hypothesis H26: *The complexity of a task affects the average hue, saturation and value of the colors used in its user interface.*

Attribute A27: Yendrikhovskij Colorfulness

Description: Colorfulness as defined by Yendrikhovskij [63]: *'The sum of the average saturation value and its standard deviation where the saturation is computed as chroma divided by lightness in the CIELab color space'.*

Rationale: Though Reinecke et al overestimated the effect of bright colors on perceived colorfulness visual complexity in their model, their analysis shows a slightly positive correlation between Yendrikhovskij's colorfulness metric and visual complexity.

Hypothesis H27: *The complexity of a task affects the Yendrikhovskij colorfulness of its user interface.*

Attribute A28: Hasler Colorfulness

Description: Colorfulness as defined by Hasler [25]: *'The weighted sum of the trigonometric length of the standard deviation in ab space and the distance of the center of gravity in ab space to the neutral axis'.*

Rationale: Similar to the rationale for attribute A24, we include this metric as a different method of measuring colorfulness.

Hypothesis H28: *The complexity of a task affects the Hasler colorfulness of its user interface.*

3.2 Attribute Evaluation

Having defined a set of task attributes to investigate, we will require a method of evaluating their relationship to task complexity. We recall our definition of task complexity from Chapter 2:
*The **complexity** of a HIT can be approximated by the sum of the complexity of its objectively measurable attributes, as perceived by its workers.*

To measure complexity, we will need two things: a **complexity measure** to approximate complexity, and the **technical means** to collect and evaluate data on its objectively measurable attributes.

3.2.1 Approximating Complexity

Deciding on a complexity measure is not straightforward. Is it something that can be objectively measured, or is it a purely subjective concept? Again, we follow Campbell's [7] framework and attempt to find a measure of complexity that is a function of objective task characteristics. With this in mind, we consider two examples of previous research linking the complexity of human intelligence tasks with their *completion time*.

In 2011, Rogstadius et al. [53] performed an assessment of different types of motivation and how these affected workers. They asked MTurk workers to identify malaria parasites and count regular blood cells in an image. They defined complexity as being a function of the amounts of blood cells and parasites: the more there were of each, the higher the complexity of the task. Comparing different motivational factors, such as the task being presented as coming from a for-profit or non-profit organization and varying the pay amount, they found that for all of these conditions the amount of time spent to complete a task increased and the accuracy of workers' answers decreased as the complexity of the task increased. However, for the highest complexity levels, the amount of time spent on the task decreased sharply, showing a demotivational effect when a task is unrealistically difficult.

In 2013, Goncalves et al. [21] performed a similar experiment, this time using public displays as the task presentation medium. Passersby had the option to interact with the display - if they did, it would present them with the blood cell categorization task. Workers whose instructions were not constructed to improve motivation spent less time on tasks than workers whose instructions did follow a motivational approach, but across all motivational conditions the trend of higher task complexity correlating with higher completion time and lower accuracy was visible.

In 2015, Difallah et al. [11] performed an experiment using several months of market data from Mechanical Turk. Based on a set of metadata attributes and market information, they aimed to predict how many HITs in a 'batch' (what we call a HIT-group) would be completed in the next hour, using market data and metadata attributes from the batch itself. They showed that several attributes (such as HITs available) could be used to effectively predict throughput for larger batches.

Based on this previous work, we propose task throughput be used as an approximation of a task's complexity for the purposes of our research. Our reasons for this are that it is objectively measurable and that the relation between the two has been observed in the previous research cited above.

3.2.2 Obtaining Task Attribute Measurements

Having decided on an objectively measurable attribute that can serve as an approximation of HIT complexity, we consider the technical means required to automate the analysis of HIT complexity. To perform this analysis, we need the following:

- **A dataset of MTurk market data.** Gathering HIT data from the largest online human computation platform should offer a sufficient quantitative as well as qualitative basis for research.
- **Tools to extract attribute measurements from MTurk data.** Given the raw market data, we should be able to input it into a pipeline that turns it into a rich dataset containing measurements for all of the attributes we identified.

We were able to work with the dataset used by Difallah et al., which was graciously provided by Gianluca Demartini. However, though the dataset offered a very large amount of data, the content of the HITs in the dataset was not stored completely: the HTML content was stored, but without external resources like script files or images, and with common stop words removed. This meant that, had we wanted to reinstantiate these tasks to measure visual attributes, they would most likely not resemble the original HITs at all due to the missing content. Measurements for some content attributes would also not be completely accurate due to the large amount of missing words.

In order to research all three categories of task attributes, this meant we needed to build our own solution to obtain HIT data from the Mechanical Turk market and measure HIT attributes. We could then use both the dataset used by Difallah et al., which has the advantages of containing a large amount of data and already having been used in a throughput prediction experiment, and our own dataset, which would offer a more complete set of features to analyze. In the next chapter, we will outline the architecture of the systems we constructed to gather our own dataset and measure the attributes of the stored tasks to make them suitable for analysis.

Chapter 4

Technical Means of Task Complexity Analysis

In this chapter, we discuss the technical means through which enabled us to carry out our analysis of task complexity. First, we discuss the tool we used to track HITs, MTurk Tracker. We describe its basic functionalities and outline the additions we made to its data retrieval process to be able to retrieve all of the data we required. After that, we discuss the structure of the pipeline we used to build a task feature dataset.

4.1 Extended MTurk Tracker

4.1.1 Introduction

MTurk Tracker [30] is an application created by Panos Ipeirotis in order to track the behavior of the MTurk HIT market. Its first iteration was created as a Python project in 2009, and it was used for several years to crawl the list of "HITs Available" on MTurk once per hour, storing data about the available HITs and their requesters. In early 2014, the project was re-imagined as a Java-based web application using the Google App Engine [1] architecture, which is still in use on the MTurk Tracker website today [2]. This GAE version of MTurk Tracker is the system we have extended for the purposes of this study. In the next two subsections, we will describe the architecture and functionalities of MTurk Tracker as well as of our own extensions, and describe in detail the data our application collected from MTurk. For clarity, we will refer to the version of the application created by Panos Ipeirotis as the 'original tracker' (OT), and to our own version as the 'extended tracker' (ET). In instances where we do not specify whether something applies to the OT or the ET, the reader should assume that it applies to both the original and the extended tracker.

4.1.2 System Overview

MTurk Tracker consists of five main components, which we term the Frontend Views, the API, the Tracking Servlets, Scheduling and the Datastore. We will describe each component's functionalities briefly.

[1] `https://cloud.google.com/appengine/`

[2] `www.mturk-tracker.com`

- **Frontend Views**: This component presents the data collected by the application to the user. These views include items such as charts showing how many HITs are posted and completed over time, or a list of top requesters by amount of HITs posted and total reward offered.
- **API**: The API [3] allows for external access of (some of) the data stored in the datastore, either through the browser or by sending HTTP GET requests.
- **Tracking Servlets**: A cluster of web servlets that are responsible from retrieving data from MTurk and storing the information in the datastore. The tracking servlets pull newly posted HITs, track the completion times for HITs that are already being tracked, and compute overall market and requester statistics.
- **Scheduling**: Utilities for scheduling the activation of the tracking servlets. The frequency of tracking for each of the servlets is defined in a cron file, and the scheduling classes manage task queues for the pulling and tracking of HITs and the computation of statistics.
- **Datastore**: MTurk Tracker uses the Google App Engine datastore, which is a schemaless NoSQL datastore. The tracking servlets store the data they retrieve in the datastore, and the datastore can then serve this data to the API and Frontend Views.

Figure 4.1 presents an overview of the architecture of the original tracker.

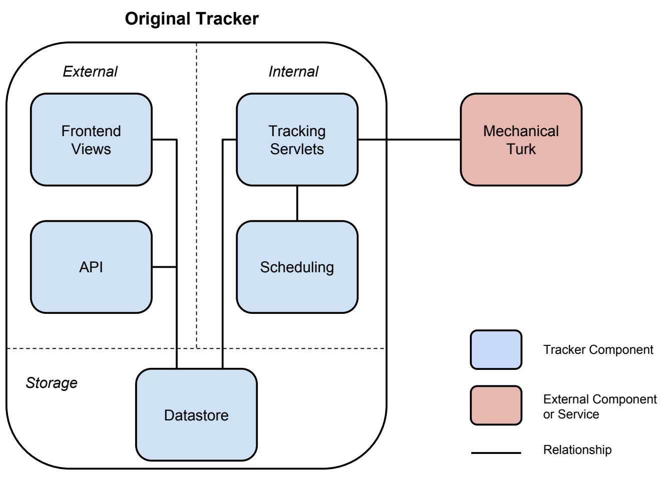


Figure 4.1: Architecture of Original Tracker

[3] `https://apis-explorer.appspot.com/apis-explorer/?base=https://crowd-power.appspot.com/_ah/api#p/mturk/v1/`

The original implementation already provides several key functionalities needed to obtain data we could use for our own research. However, there were a few key concerns that motivated us to extend the tracker rather than use the original.

- Due to the way the original tracker implemented API requests, it was not possible to retrieve the full contents of the datastore using its API. This was a factor in our decision to collect our own dataset, rather than retrieve data from the original tracker.

- While the original tracker stored HTML content of HITs, it stored the content with commonly occurring words removed and without storing additional content like CSS, JavaScript and image files. Since we were interested in restoring previously posted HITs as accurately as possible, we would need to add this functionality ourselves.

Figure 4.2 shows the additions we made to the system. The code of the original tracker was changed to allow it to send HTTP requests to our external server, reflected in the *Request Tools* component and the changes in the Tracking Servlets component, denoted by an asterisk. The Tracking Servlets component was also modified to call this component when new HITs are inserted into the datastore. The content extraction server hosted separately from the tracker, which we shall call the content retriever, features the *Servlets* and *Extractor* components. The Servlets component accepts HTTP POST requests, and the Extractor component allows for the automated extraction and storage of content from HTML pages. The external libraries that were used to develop the additional functionality include Apache Tomcat 7 for web app architecture, JSoup for HTML document processing, and several Apache Commons libraries for various purposes including file operations and text validation. For project management, quality assurance and testing, the following tools were used: Apache Maven, Cobertura, PMD, FindBugs, CheckStyle, JUnit, Mockito and PowerMock.

The main additions to the tracking process by the extended tracker can be summarized as follows:

1. After the tracker stores a new HIT group in the datastore, the tracker sends the groupID and original URL of the HIT group to the content retriever via a HTTP POST request.

2. The content retriever retrieves the web page found at the URL and stores it locally, renaming it to {groupID}.htm(l).

3. The content retriever identifies any CSS, JavaScript and images included in the page.

4. The content retriever downloads the identified content and stores it in the same folder as the retrieved page.

5. The content retriever changes all links to external content in the retrieved page so that they point to the content stored in its own folder.

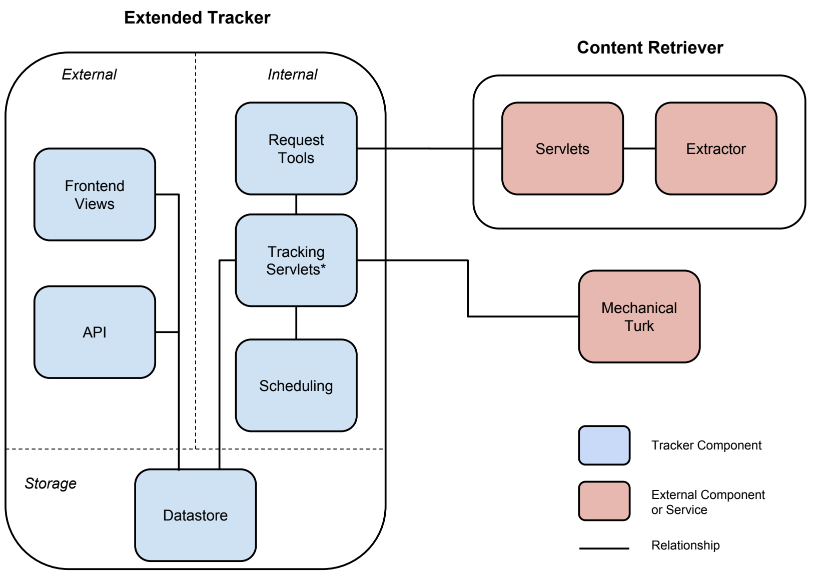
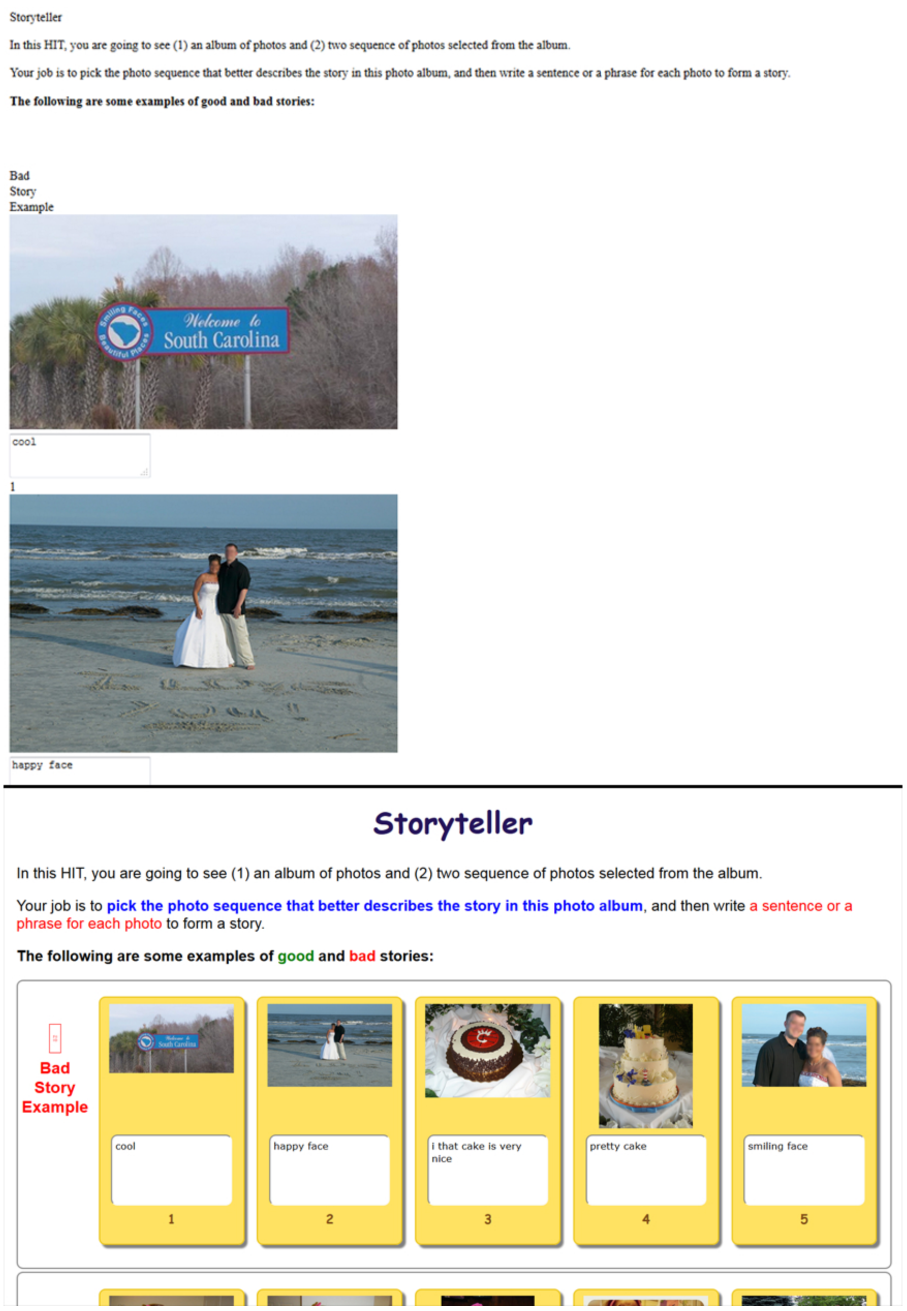


Figure 4.2: Architecture of Extended Tracker

6. The restored HIT can now be viewed directly, identifiable by its *groupID*.

Because of these additions, it is possible to view tracked HITs in a restored state that includes their original style, script and image files. The process is not perfect: retrieving external resources might fail, or the page might still not render correctly due to formatting issues in the substitution. However, we did not encounter any serious issues in working with restored HITs and are of the opinion that HITs restored using the extended tracker will always provide a representation of the original HIT that is at least as accurate as a representation constructed using data from the original tracker, and in most cases will outperform the latter when evaluating how similar the restored HIT is to the original HIT.

Figures 4.4 and 4.3 show comparisons between HITs as stored by the original tracker and as restored by our extended tracker. In Figure 4.4, the differences are relatively subtle. While the original tracker provides a reasonable restored HIT, the extended tracker restores the interface more accurately, showing more of the originally intended functionality and layout. In Figure 4.3, the difference is much more noticeable: the original tracker did not preserve the intended layout or functionality at all, while the extended tracker provides a very accurate depiction of the original HIT. In addition to the immediately visible improvements to the interface, our tracker also has the advantage of storing images and script files locally: this means it will not be a problem if these files are no longer available from the original source when a HIT is restored.



Storyteller

In this HIT, you are going to see (1) an album of photos and (2) two sequence of photos selected from the album.

Your job is to pick the photo sequence that better describes the story in this photo album, and then write a sentence or a phrase for each photo to form a story.

The following are some examples of good and bad stories:

Bad
Story
Example

cool

1

happy face

Storyteller

In this HIT, you are going to see (1) an album of photos and (2) two sequence of photos selected from the album.

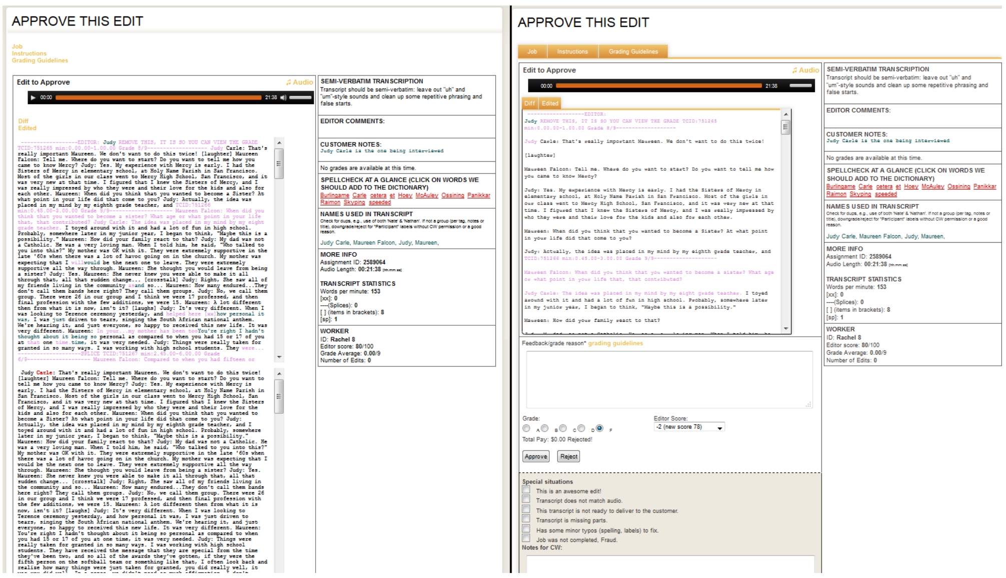
Your job is to **pick the photo sequence that better describes the story in this photo album**, and then write a sentence or a phrase for each photo to form a story.

The following are some examples of good and bad stories:

Bad Story Example

cool	happy face	i that cake is very nice	pretty cake	smiling face
1	2	3	4	5

Figure 4.3: Restored HIT: Image Annotation.
Original tracker version above, extended tracker version below.



SEMI-VERBATIM TRANSCRIPTION
Transcript should be semi-verbatim: leave out "uh" and "um"-style sounds and clean up some repetitive phrasing and false starts.

EDITOR COMMENTS:

CUSTOMER NOTES:
Judy Carle is the one being interviewed

No grades are available at this time.

SPELLCHECK AT A GLANCE (CLICK ON WORDS WE SHOULD ADD TO THE DICTIONARY)
Burlingame Carle cetera et Hoey McAuley Ossining Panikkar Raimon Skyping speeded

NAMES USED IN TRANSCRIPT
Check for dups, e.g., use of both 'Nate' & 'Nathan'. If not a group (per tag, notes or title), downgrade/reject for "Participant" labels without CW permission or a good reason.

Judy Carle, Maureen Falcon, Judy, Maureen,

MORE INFO
Assignment ID: 2589064
Audio Length: 00:21:38 [hh:mm:ss]

TRANSCRIPT STATISTICS
Words per minute: 153
[xx]: 0
----(Splices): 0
[ ] (items in brackets): 8
[sp]: 1

WORKER
ID: Rachel 8
Editor score: 80/100
Grade Average: 0.00/9
Number of Edits: 0

```
Maureen: When did you think that you wanted to become a Sister? At what point
in your life did that come to you?

Judy: Actually, the idea was placed in my mind by my eighth grade teacher, and
TCID:751266 min:0.45.00-3.00.00 Grade 9/9--------------------

Maureen Falcon: When did you think that you wanted to become a sister? What age
or what point in your life that, that contributed?

Judy Carle: The idea was placed in my mind by my eight grade teacher. I toyed
around with it and had a lot of fun in high school. Probably, somewhere later
in my junior year, I began to think, "Maybe this is a possibility."

Maureen: How did your family react to that?
```

Feedback/grade reason* grading guidelines

Grade: A B C D [✓ F]
Editor Score: -2 (new score 78)
Total Pay: $0.00 Rejected!

Approve Reject

Special situations
- [ ] This is an awesome edit!
- [ ] Transcript does not match audio.
- [ ] This transcript is not ready to deliver to the customer.
- [ ] Transcript is missing parts.
- [ ] Has some minor typos (spelling, labels) to fix.
- [ ] Job was not completed, Fraud.

Notes for CW:

Figure 4.4: Restored HIT: Transcription Checking.
Original tracker version on the left, extended tracker version on the right.

4.1.3 Stored Data

Table 4.1 shows the structure of the data that is stored as implemented in the original tracker. The datastore is not relational, but the *ID/Name* field serves to identify entries in the HITgroup, HITinstance and HITcontent tables as belonging to a single HIT group, the *requesterID* field in HITgroup corresponds to the ID/Name in HITrequester, and the *requesterName* field links HITrequester and TopRequester. Initially, there were two fields that did not function as intended in the original tracker: both *projects* in TopRequester and *qualifications* in HITgroup did not collect any data during our trial runs. We believe this is due to projects no longer being supported by MTurk and thus not being retrievable, and the code for retrieving and storing qualifications simply containing bugs. Eventually, the original tracker was updated to fix the storing of qualifications. We incorporated these changes into the extended tracker as well, and as far as we are aware storing qualifications functioned correctly during data collection.

We tracked the MTurk market using our extended tracker from August 3rd, 2015 at 10:00AM through midday on August 11th, 2015. This means we had more than a week of market data. Overall, we tracked and stored metadata for 5487 HITgroups. For all but 81 of these HITgroups, we were also able to store the HTML content. The combined storage required for the HITgroup content was 6.2GB uncompressed and 1.7GB compressed in a ZIP archive. Our tracking frequency was the same as Ipeirotis' [30] - we tracked new and existing HITs every 15 minutes. We believe that this was an adequate interval, as the data gathered by Ipeirotis suggested that HIT groups typically have a completion time between several hours to several days.

Table Name	Fields	Description
ArrivalCompletions - Aggregates data from MarketStatistics to provide market statistics for certain periods of time	*ID/Name*	Identifier
	from	Timestamp of start of the measurement interval
	to	Timestamp of end of the measurement interval
	hitGroupsArrived	Amount of newly posted HITgroups in interval
	hitGroupsAvailableUI	Amount of available HITgroups in interval as visible on MTurk dashboard
	hitGroupsCompleted	Amount of completed HITgroups in interval
	hitsArrived	Amount of newly posted HITs in interval
	hitsAvailableUI	Amount of available HITs in interval
	hitsCompleted	Amount of completed HITs in interval as visible on MTurk dashboard
	length	Length of the time interval in minutes
	rewardsArrived	Amount of rewards added in the interval in cents
	rewardsCompleted	Amount of rewards for completed tasks in interval in cents
MarketStatistics - Market statistics at certain points in time.	*hitGroupsAvailable*	Amount of available HITgroups for given timestamp
	hitsAvailable	Amount of available HITs for given timestamp
	timestamp	Timestamp of measurement
HITgroup - Represents a group of HITinstances.	*ID/Name*	Identifier, matches HIT group ID on MTurk website
	active	Whether or not the HITgroup is currently active
	description	Short description of the task given by the requester
	expirationDate	Timestamp for the time at which the HITgroup will automatically expire
	firstSeen	Timestamp for the first time this HITgroup was found to be active by the tracker
	keywords	Keywords for the task given by the requester
	lastSeen	Timestamp for the most recent time this HITgroup was found to be active by the tracker
	qualifications	Qualifications required to complete the task - in quotes, separated by commas
	requesterId	Requester identifier, matches requester ID on MTurk website
	reward	Reward listed in the task description
	timeAllotted	Time allotted to complete the task in seconds
	title	Title of the HITgroup in task description
HITinstance - represents a single HIT instance at a single point in time. HITinstances are part of HITgroups and can contain one or more HITs.	*ID/Name*	Identifier, made up of a unique HITinstance identifier appended to the groupId
	groupId	groupID of the HITgroup this instance belongs to
	hitsAvailable	Amount of HITs in this instance available for completion
	hitsDiff	Difference in hitsAvailable since last time this was tracked. Positive if this is the first measurement of this HITinstance, zero or negative otherwise
	rewardDiff	Difference in rewards available since last time this was tracked. Positive if this is the first measurement of this HITinstance, zero or negative otherwise
	rewardsAvailable	Remaining reward available for completing HITs in this batch
	timestamp	Timestamp for this measurement
HITcontent - represents the HTML content of a HIT	*ID/Name*	Identifier, matches HIT group ID on MTurk website
	content	HTML content of the HITgroup, or null if HIT is not externally hosted or content could not be retrieved.
	url	Original URL at which the HITcontent was hosted. Null if HIT was not externally hosted.
HITrequester - represents a party offering HITs	*ID/Name*	Requester identifier, matches requester ID on MTurk website
	lastActivity	Timestamp for last detected activity
	requesterName	Requester name as displayed to workers
TopRequester - measures requester statistics for each day	*ID/Name*	Identifier, made up of requesterId appended to the Unix epoch timestamp for the relevant day
	hits	HITs posted by the requester that day
	projects	Unused field (always 0)
	requesterId	Requester identifier, matches requester ID on MTurk website
	requesterName	Requester name as displayed to workers
	reward	Total reward provided by requester that day
	timestamp	Timestamp for the date

Table 4.1: Tables and Fields in MTurk Tracker Datastore

4.2 Task Attribute Analysis Pipeline

4.2.1 Data Sources

In order to process the market data gathered by the extended tracker, we constructed a pipeline for automatically measuring attributes. We refer to this pipeline as the HITProcessor. Figure 4.5 provides an overview of how data is gathered from different sources and presented to the HITProcessor. This process can be summarized as follows. The CSV parsing and writing referred to in this section was performed using the *opencsv* [4] package for Java. This subsection specifically discusses processing the data gathered from our own tracker - we made minimal adjustments to also process the dataset used by Difallah et al., which was provided as an SQL database. The attribute calculation process described in the next subsection is identical for both types of data.

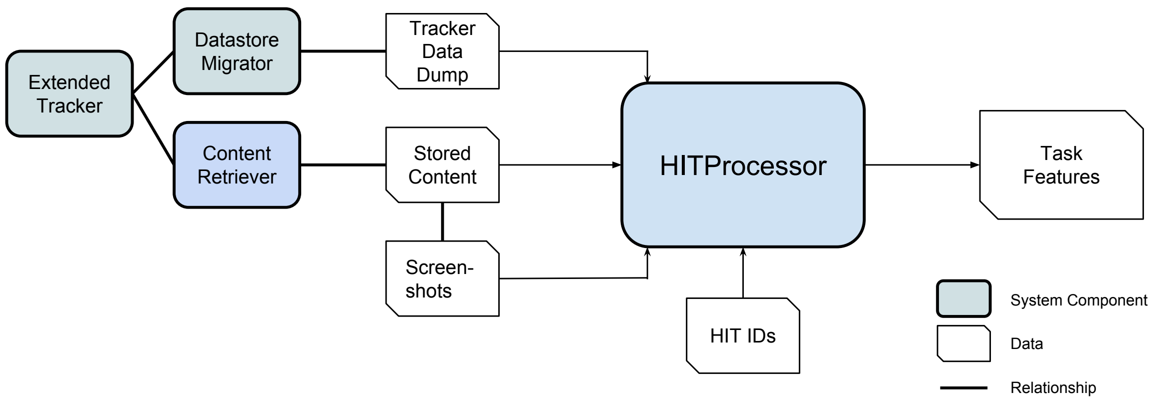


Figure 4.5: Task Attribute Analysis Pipeline - Overview of Data Sources

1. The **Extended Tracker** tracks HITs and stores HIT metadata. It sends requests to the **Content Retriever**, which downloads and stores HIT content files.

2. The data stored on the tracker is exported as a **Tracker Data Dump** in CSV format using the **Datastore Migrator**.

3. The **Restored Content** is archived and downloaded from the Content Retriever.

4. **Screenshots** are made of HITs from the Restored Content and stored locally.

5. The **Tracker Data Dump, Stored Content** and **Screenshots** are provided to the **HITProcessor** as input.

6. Using the data, the HITProcessor generates a **Task Features** CSV file containing task attribute measurements for the HITs listed in the provided **HIT IDs** file.

[4] http://opencsv.sourceforge.net/

After the task features CSV has been generated, it can be used for analysis. In order to generate this file, the HITProcessor goes through four distinct phases. We describe each phase and how the features in each category are calculated below.

4.2.2 Attribute Calculation

In this section, we describe the process of attribute calculation for a single HIT group as performed by the HITProcessor.

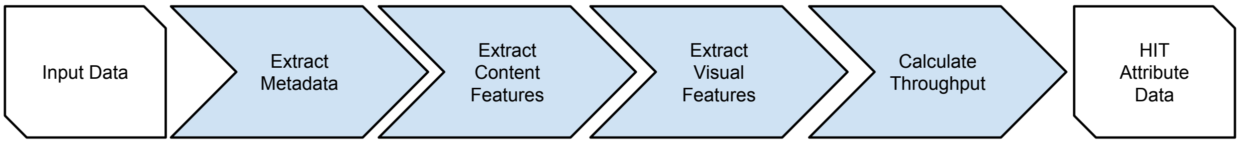


Figure 4.6: Task Attribute Analysis Pipeline - Attribute Calculation Process

Given the input data described in the previous subsection, the first phase of the process is **Metadata Extraction**. From the HITgroup table found in the data dump from the tracker, the system reads metadata attributes *A1* through *A9*. These are available either directly from the data (reward) or require some effort to retrieve. For example, qualifications attributes *A3, A4, A5* and *A6* are generated by searching the qualifications string for each given qualification type.

The second phase of the process is **Content Feature Extraction**. By processing the stored HTML page with *jsoup* [5], content attributes *A11* through *A21* are measured and stored.

The third phase of the process is **Visual Content Extraction**. In this phase, we use the tools developed by Reinecke et al. [52] for their own work. In short, this functions as follows:

1. The HITProcessor locates the screenshot corresponding to the ID of the current HIT.
2. The screenshot is decomposed into visual 'Blocks' in a tree structure, and these are represented using an XML file.
3. The XML file is analyzed to calculate features *A22, A23* and *A24*.
4. The color distribution of the screenshot is analyzed to calculate features *A25* through *A28*.

For a more detailed explanation of the feature calculation, we refer to the paper by Reinecke et al. and their website [6], which provides the source code for calculating the

[5] `http://jsoup.org/`

[6] `http://iis.seas.harvard.edu/resources/aesthetics-chi13/`

visual features.

The fourth phase of the process is **Throughput Calculation**. Using the HITinstance and ArrivalCompletions tables from the tracker data, we calculate attribute *A10, hits available*. Our measure for throughput is given for a HITgroup H at a point in time t.

$$throughput = hitsCompleted(H, [t-1, t)\ )$$

That is, the throughput for HITgroup H at time t is the amount of HITs completed between time $t-1$ until (but not including) time t. Though we tracked HIT completions every fifteen minutes, we aggregated this data to calculate throughput per hour, so that our data would be as similar to the dataset used by Difallah et al. as possible. Thus, for every HITgroup, the tracker could output its attribute features (which are the same at every timestamp) and its features that changed over time (HITs available, HITs completed, rewards paid out, et cetera).

Initially, we also experimented with a different definition for the throughput of a HITgroup:

$$throughput = \frac{amount\ of\ completed\ HITs}{HITgroup\ uptime}$$

Using this calculation, we can estimate the average *completion time per HIT* of a HITgroup within the time it was available to workers. This is what we used as an approximation of complexity, as described in the previous chapter. Unfortunately, the initial results of throughput prediction experiments using this definition were not promising: the prediction was very inaccurate and skewed greatly by a small amount of HITgroups that had much larger batches than most other HITgroups in the test set. Even with these HITs removed, however, the results still left a lot to be desired, and therefore we decided to use only the first definition of throughput given above - the one also used by Difallah et al.

Chapter 5

Evaluation of Results

In this chapter, we evaluate the data we obtained in several ways. First, we provide general observations on the data. Then, we analyze the distributions and relationships of the various attributes identified in Chapter 3. Second, we discuss our re-enactment of the throughput prediction experiment performed by Difallah et al. incorporating our own task attribute features. Third, we present an experiment to measure subjective complexity using a set of reinstantiated tasks. Finally, we summarize our findings in the form of a set of recommendations for human computation task designers.

5.1 Overview of Data

We tracked the Mechanical Turk market using our extended tracker between August 3rd and 11th, 2015. In this time, we tracked a total of 8,280 HITgroups representing 562.165 HITs being posted on the market. Our sample contained 5,487 HITgroups, or approximately 66 percent of all HITgroups. Keeping in mind there might have been temporary outages on the part of our tracker or the MTurk website, or other issues might have occurred during tracking, this figure seems to match with our earlier estimation that slightly over two-thirds of the market consists of externally hosted (and thus 'trackable') HITgroups. In our sample, we observe nearly 90 percent of the HITgroups were uploaded by one of only three requesters, as shown in Table 5.1.

Requester	**HITgroups**	**Percentage**
SpeechPad	2953	53.82%
Bunny Inc	1118	20.38%
CastingWords	842	15.35%
Others	574	10.46%

Table 5.1: Distribution of Requesters by amount of HITgroups

SpeechPad and CastingWords offer transcription HITs, while Bunny Inc. offers audio annotation HITs. These 'corporate' requesters dominate the market with thousands of HITs, while the other 154 requesters posted only 574 HITgroups all together. Difallah et al. also noted that transcription and audio annotation tasks were among the fastest growing market segment in their dataset. We also note that HITs by a single requester typically have the same objective and interface. Analyzing visual features for the entire market would therefore skew the results massively in favor of the top three requesters, whose HITs always use the same layout. In addition to this, many of the HITs posted by smaller requesters are unsuitable for reinstantiation, and if we cannot restore a HIT accurately, we cannot accurately analyze its visual attributes, either. There are several reasons why a HITgroup might not be suitable for visual feature analysis:

- **Missing Content:** The HIT included content the tracker was unable to store, but is necessary to accurately reinstantiate the HIT. This could include a Flash browser game or other types of server-side resources.
- **Not a HIT:** The actual HIT is password-protected to allow only certain workers to complete it, or is simply a link to another website where the actual HIT (typically a survey) is performed.
- **Cannot Instantiate:** Either the requester uploaded a HIT that does not function properly, or our system did not store it correctly. For 81 out of 5,487 HITgroups (1.4%), the tracker did not download any content at all. Other HITs in this category might only show a blank screen or contain severe bugs that make reconstruction impossible, either due to an error on the part of the requester or due to incomplete storage.

In order to obtain a dataset that would represent a diverse sample of actual HITs workers would be able to complete, we manually investigated restored HITs uploaded by all 157 requesters. In the end, this resulted in a dataset of 76 HITgroups, each from a different requester, that we considered reinstantiable and representative of actual HITs. While we could have included all HITs uploaded by these requesters, we observed that requesters typically upload HITgroups that are nearly identical, and thus included only a single HITgroup per requester to ensure our sampling of HITgroups was as diverse as possible. We took screenshots of these tasks to be used for visual feature analysis.

Task Type	**Count**	**Percentage**
Survey (SU)	4	5.26%
Verification and Validation (VV)	2	2.63%
Interpretation and Analysis (IA)	23	30.26%
Content Creation (CC)	25	32.89%
Information Finding (IF)	14	18.42%
Content Access (CA)	4	5.26%
Other	4	5.26%

Table 5.2: Distribution of Task Types in Filtered Dataset

We also analyzed the task types of the selected HITgroups, following the classification used by Gadiraju et al. [20] that was also used by Difallah et al. The results can be seen in Table 5.2: all task types are represented, though some are more numerous than others. The three task types that our dataset contains most - IA, CC and IF - were also the most popular HIT types in the dataset used by Difallah et al, along with SU. We attribute this difference to the fact that surveys are commonly posted as a 'redirect' HIT: users complete a survey on an external site, and return to MTurk to fill out a completion code and receive their payment. This meant that these types of HITs did not qualify for our dataset, as we could not reinstantiate them.

5.2 Task Attribute Analysis

In this section, we will discuss our findings with regards to the different types of attributes. We discuss them divided into the three categories we defined in Chapter 3: Metadata, Content and Visual. We specifically look at our filtered dataset to ensure a diverse sample: if we were to perform this analysis for a large dataset without filtering, the numbers would most likely be skewed by the large amounts of near identical HITs from the small amount of top requesters. Figure 5.1 shows a correlation matrix for the tracked attributes in the one-week dataset, to illustrate the correlations between the various attributes. It does not include the semantic attributes or throughput, as these are multidimensional and as such do not lend themselves to being plotted in such a way. The table shows the level of correlation between pairs of attributes: a light to dark blue square indicates an increasingly negative correlation, while an orange to red square indicates an increasingly positive correlation. We will refer back to this table in the next subsections when we discuss the attributes.

5.2.1 Metadata Attributes

First, we consider **Metadata** attributes.

Rewards offered varied between $0.00 and $8.00. Not considering the single $8 outlier, the average reward was $0.19. The average time allotted to complete a HIT was slightly under 2 hours - requesters typically allot much more time than a HIT actually takes to complete.

As for *qualifications*, 10 (13%) required the worker to have a certain geographical location, only 1 (1.3%) required a Master qualification, 3 (3.9%) required a minimum amount of approved HITs, and 11 (14,5%) required a minimum approval rate, varying between 50% and 98%. We note a slight correlation between the different qualification types (apart from Master) - it appears that when a requester adds qualifications to a task, they tend to add more than one. Especially the correlation between *total approved* and *approval rate* is clearly present: when requesters require a minimum amount of approved HITs, they tend to also require a certain success ratio for those HITs, and vice versa. The virtually nonexistent use of Master qualifications is interesting: perhaps requesters feel the additional fee required to hire Master workers is not worth it. Adding qualifications to a HIT implies making a trade-off between speed and quality: by limiting the amount of eligible workers, the work will not be completed as quickly, but workers with better qualifications can be expected to perform better. We observe that in most cases, requesters choose speed over quality - this might also be due to the fact that requesters believe the quality delivered by non-qualified workers is not significantly lower than those by qualified ones, or they might already be taking their own quality control measures such as including honeypot questions or manually

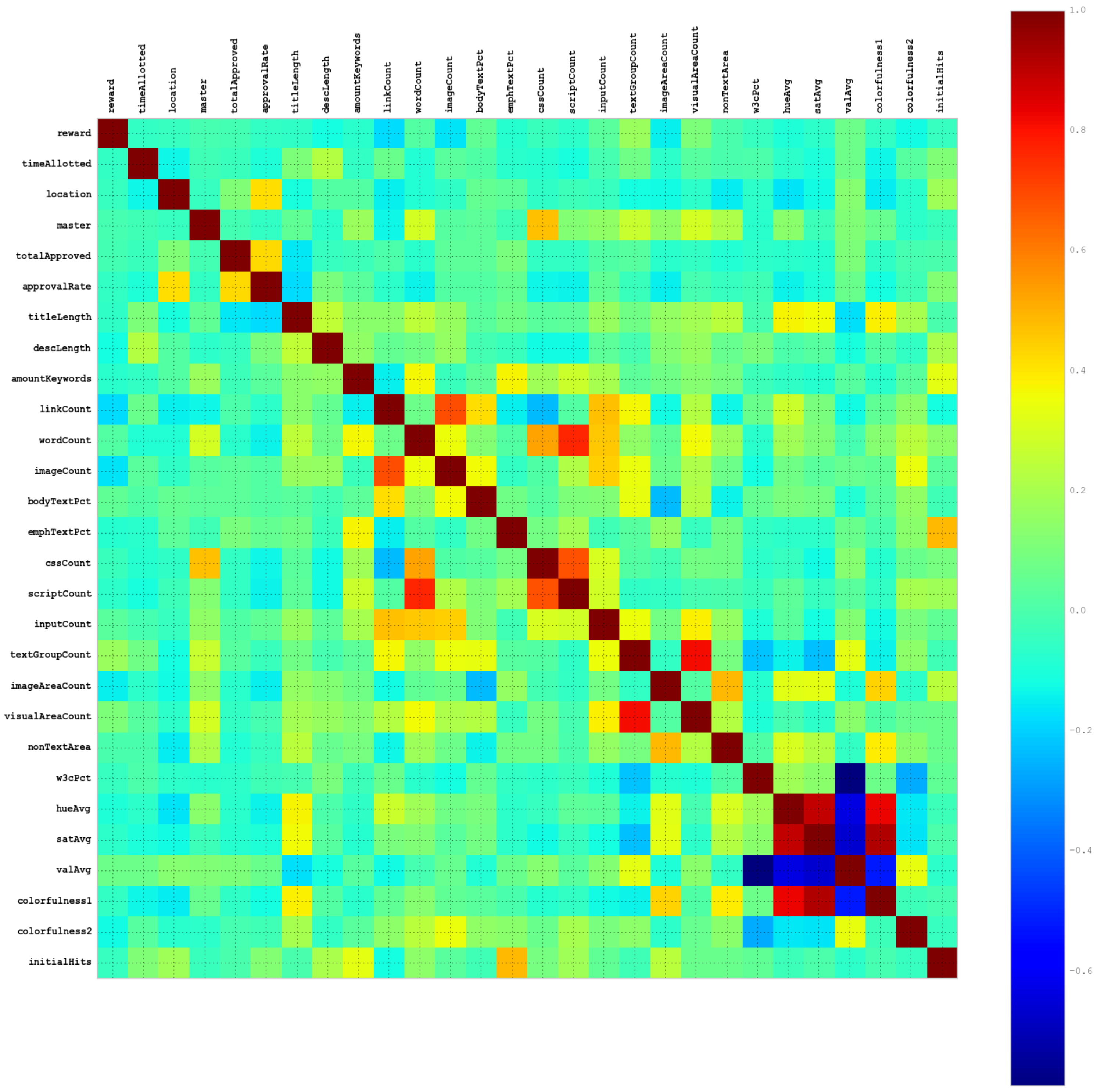


Figure 5.1: Task Attribute Correlation Matrix

checking responses.

The distributions for *title and description length* in Figures 5.2 and 5.3 look very similar. It seems task creators realise there is a 'sweet spot' when it comes to describing your task: in these cases, that spot is between 30 and 40 characters for a title and between 30 and 60 characters for a description. We observe that the actual description of a task is usually already done in the title, with the description typically not adding a lot of information: for example, the HIT with title *"Transcribe Field Text (Handwriting or Machine Print) "* has the following description: *"The task is to review the handwritten or printed text on the image and transcribe it*

sic

content." Since many requesters follow this pattern, it makes sense that the average description length is only slightly higher than the average title length.

5.2.2 Content Attributes

Next, we consider the **Content** attributes. Figure 5.4 shows the distributions for the amounts of links, words, images, CSS files, JS files and input elements on HIT webpages.

Starting with *link count*, we observe pages often contain a lot of hyperlinks. This is partly due to a difference in the way external HITs are stored: some are entirely their own website, while others are captured with a part of the MTurk interface surrounding them. This interface still contains a lot of links. We believe that if these links were not included in the link count, the average would be much lower. We also observe a positive correlation between link count and image count. This is due to images sometimes being used as links: for example, a custom button might be an image that doubles as a link to the next page.

Word count is also interesting: while there are outliers, most HITs have a rather low word count. This makes sense, as requesters would want to make their instructions as straightforward as possible. Word count shows positive correlations with image, CSS, script and input counts, as can be seen in Figure 5.1. We explain this by noting that more functionality and content in a HIT will typically require more text: if the HIT has a custom user interface, indicated by a high amount of script and stylesheet files, we can expect it to require more instructions than a very basic HIT without scripts. Additional images might require explanations per image as well, and more input counts automatically lead to more words because every input element needs a value, which contributes directly to the word count.

As for *scripting*, we observe that most HITs (49 out of 76) use a single CSS file, and a similar amount (47 out of 76) use six JS files. The most commonly used libraries include Bootstrap, jQuery and Node.js. The subset of HITs stored within the MTurk interface likely increases the average amount of scripts in this case as well. In the correlation matrix, we observe a strong correlation between stylesheet and script count, which is no big surprise. Stylesheets and scripts go hand in hand in web design: when a requester wants a HIT with specific functionalities, higher numbers of both

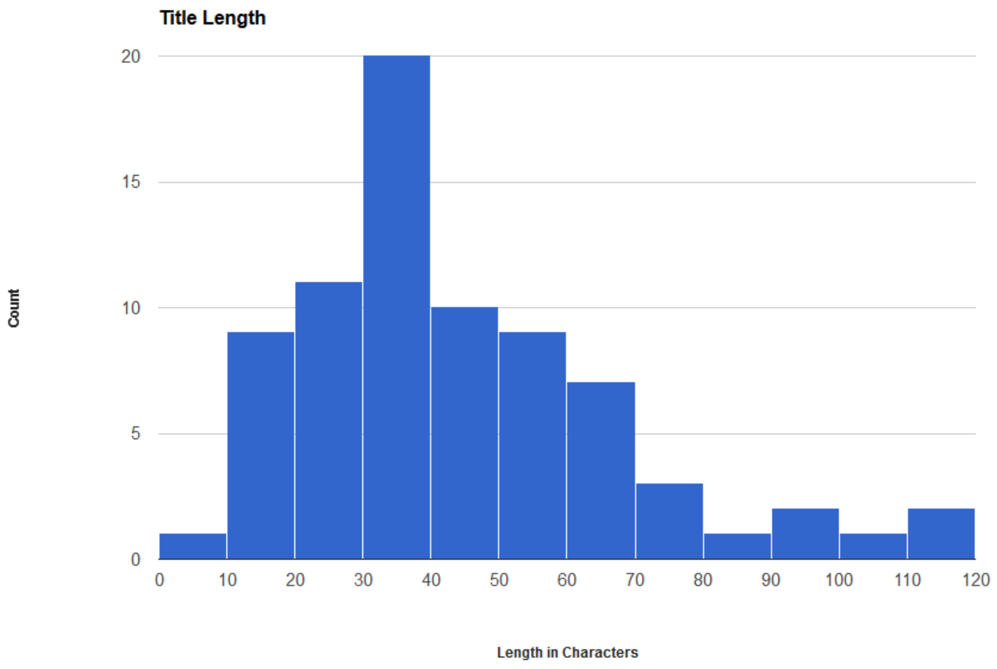


Figure 5.2: Title Length Distribution

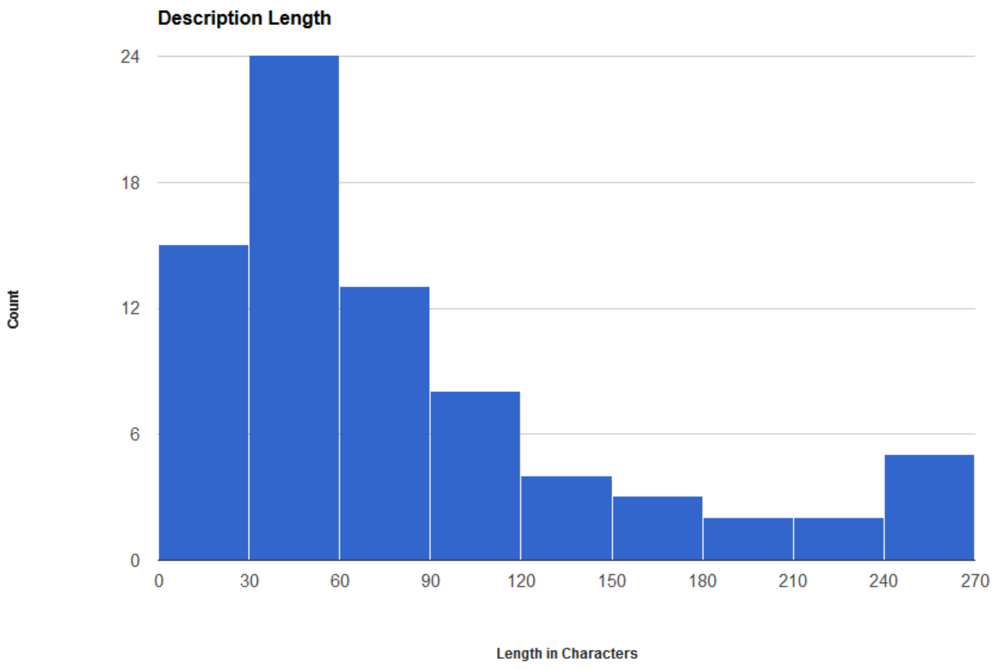


Figure 5.3: Description Length Distribution

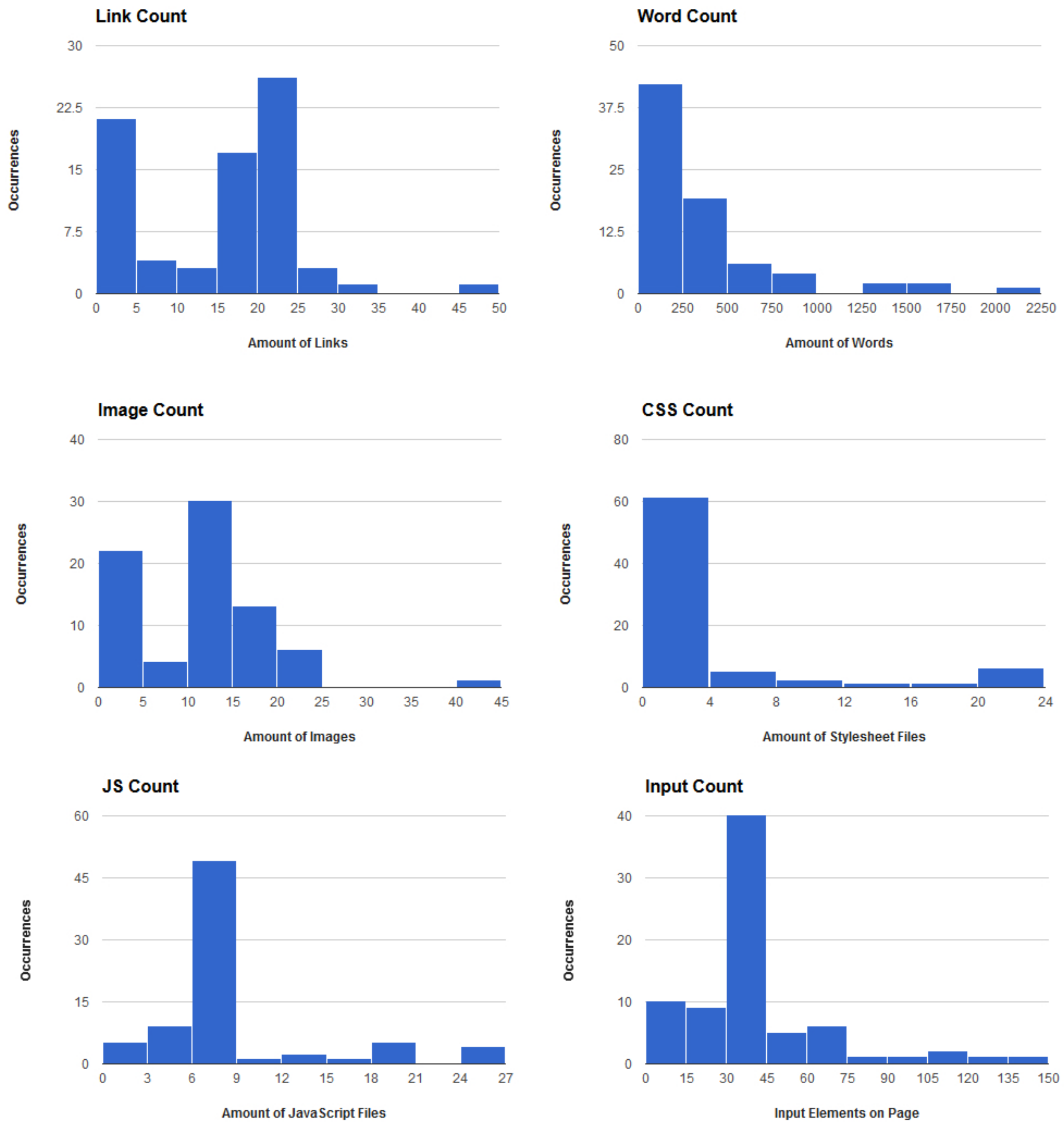


Figure 5.4: Distributions of Several HIT Content Attributes

stylesheets and scripts will be required to implement more desired features.

Regarding *input count*, we observe that HITs typically have between 30 and 45 input elements, and sometimes even many more. This is due to the fact that forms are the main tool used by requesters to obtain data from workers, and are also used for various purposes. For example, a HIT might consist of a demographic survey, an image annotation task and a small questionnaire on the quality of the HIT, all three of which can be expected to use HTML input elements to collect information.

Finally, we have no notable observations regarding the *body text percentage* and *emphasized body text* attributes. The body text percentage attribute was invariably close to or equal to 100 percent, making it impossible to meaningfully distinguish between different entries. In addition, the emphasized body text attribute showed no meaningful correlations.

5.2.3 Visual Attributes

Finally, we look at the **Visual** attributes, which we only measured in our own dataset. As we observed when discussing the content features, the presence of the MTurk interface in some HITgroups changes the results of the attribute measurements due to the added content. We believe there are two possible ways to view this: the MTurk wrapper can be viewed as a part of the HIT, or not. The argument for including the MTurk wrapper is that since we are interested in visual complexity, we should include all visual aspects of a page in the measurement, even those not directly relating to the task. The argument for omitting the MTurk interface would be that it does not relate to the task itself, which in these cases usually has a very simple layout. Thus, omitting the MTurk interface might allow the measurements to reflect the complexity of the task itself more accurately. In our own research, we adopted the first position, but we also conducted measurements on our dataset with the MTurk interface omitted from the relevant HIT groups to check if there were any noticeable differences.

Because a large part of the page was removed from a large amount of HITs, the values for several attributes (both in the Content and Visual categories) were different. However, both the distributions of and correlations between the changed attributes did not show any unexpected results. As for the distributions, it is to be expected that when a part of the page is removed, values like the amount of visual areas or the amount of images on the page decrease. This naturally caused a shift in several attribute distributions, but we did not observe any other effects. Additionally, the lower values on average means that the relative difference between attribute values becomes larger for these attributes. This could possibly lead to more pronounced correlations between attributes, which we also observed in a few cases. However, this change was invariably very small, and as such we do not consider these cases significant.

Going back to the full dataset without the omission of the MTurk interface, Figure 5.5 shows the distributions for several visual attributes. The amount of text groups was distributed relatively uniformly between 7 and 218 with a spike in the range between 100 and 125, and correlated positively with the link count, image count, input

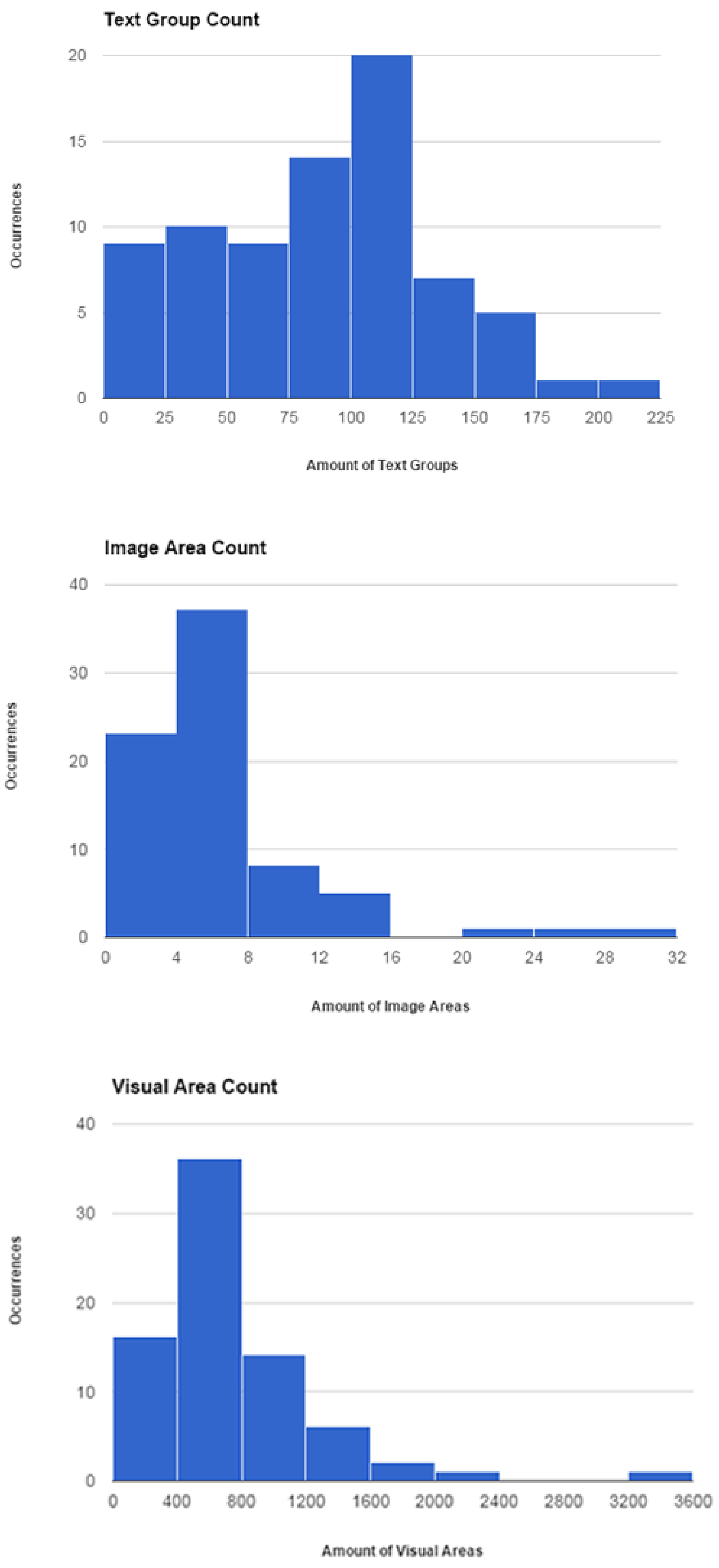


Figure 5.5: Distribution of Visual Attributes

count and the percentage of body text. It seems likely that when text is 'interrupted' by images, links, input elements and headers, it is visually split into distinct groups. Moving on, the amount of image areas correlated slightly with Yendrikhovskij colorfulness, but did not correlate with the amount of images. In fact, the amount of image areas is typically smaller than the amount of images for a given HITgroup. Perhaps this is due to the fact that the distribution of images on pages (spread out or cluttered) varies too much between HITs for there to be a correlation between image count and visual image areas, since several images next to each other are considered to be a single image area.

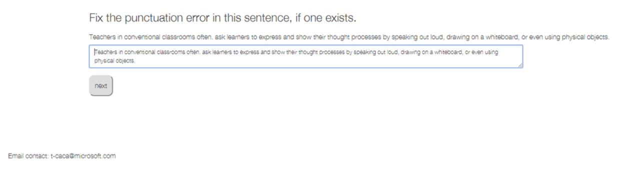
For the amount of visual areas, we observe weak positive correlations with word count and input count and a strong correlation with image count. This leads to the hypothesis that as a HIT grows in size, it becomes more visually complex. To illustrate, Figure 5.6 shows the HITgroup from our sample with the highest amount of visual areas (over 5,000) and a HITgroup with a small amount of visual areas (fewer than 200). The layout of the HIT with few visual areas is very simple, while the larger HIT is much more visually complex. While a HIT in which more questions are asked will necessarily be larger than one with few questions, requesters should consider how they present each question visually to minimize the risk of information overload.

Regarding colorfulness, we see clear relationships between the average HSV values of a page and Hasler Colorfulness. Though this tells us more about the way Hasler Colorfulness is computed, it does not imply insights about HIT colorfulness. Yendrikhovskij colorfulness does not show any noteworthy correlations with other attributes, either. Based on our own observations of the dataset, we note that HITs do not tend to be colorful. While some HITs include images, most HITs we evaluated consisted of mostly black text against a white background with a few grey input elements. While this is not a negative attribute, per se, it does mean that assessing HITs according to visual attributes, especially color-related attributes, will most likely require further refinement beyond what we have done in this work.

5.3 Throughput Prediction

5.3.1 Setup

In our own throughput prediction experiment, we attempted to follow the setup used by Difallah et al as closely as possible. From their dataset, we took hourly observations of the MTurk market from June to October 2014, and uniformly sampled 50 test points to be evaluated. For our own dataset, we used observations of every suitable hour in our time range of one week, suitable timeframes being hours in which HITs in our dataset were being completed. The construction of the training and test sets for the prediction algorithm is depicted in Figure 5.7. The goal is to predict throughput for any HITgroup active at time T. Throughput is defined as the amount of HITs in that group that will be completed in the hour following time T - this data is marked as the 'test set' in the image. We built a training set for the prediction using samples from all HITs on the market for the time range $[T-\delta, T)$. Difallah et al. found the best results for $\delta = 4$,

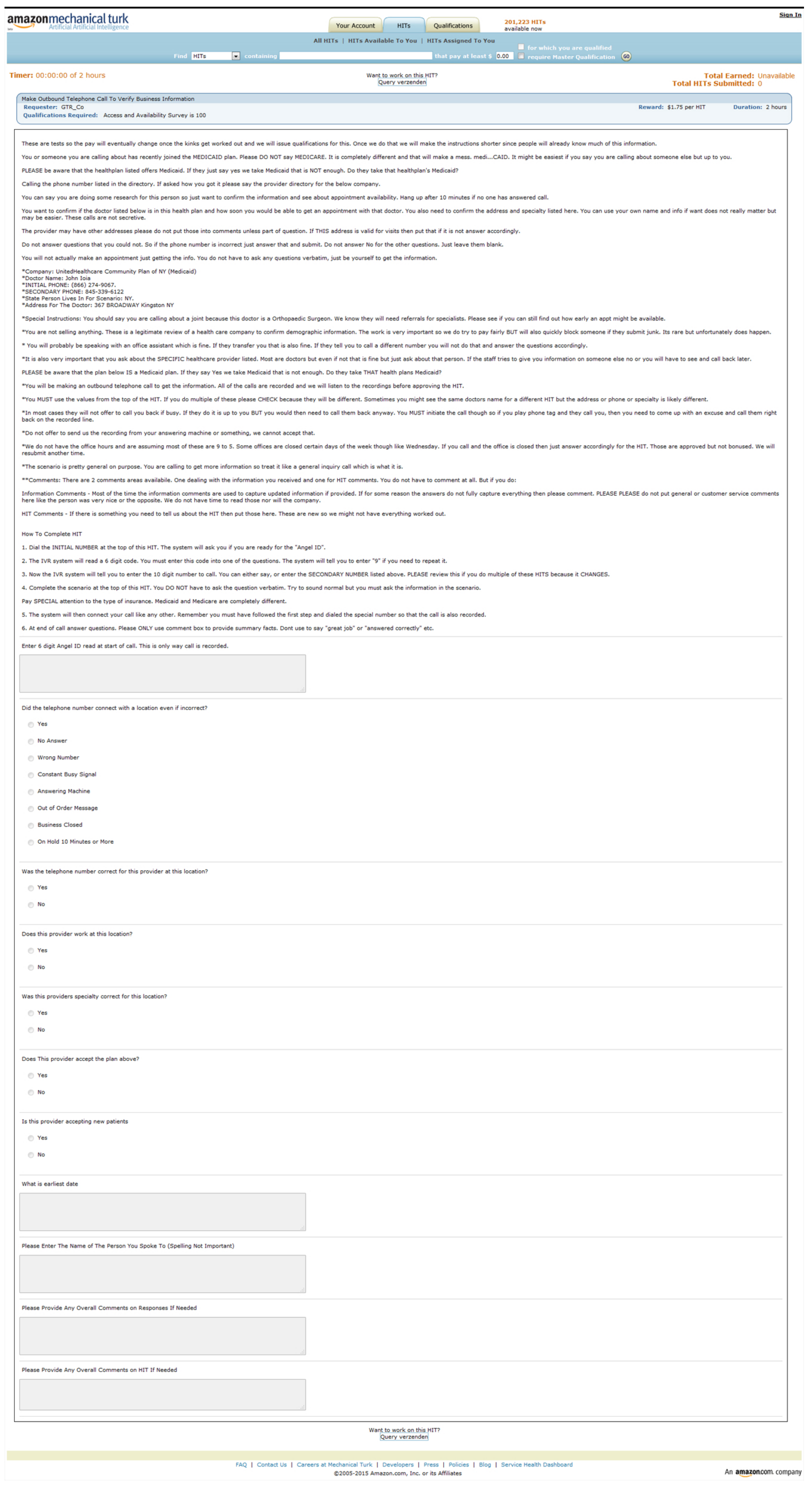


Fix the punctuation error in this sentence, if one exists.

Teachers in conventional classrooms often. ask learners to express and show their thought processes by speaking out loud, drawing on a whiteboard, or even using physical objects.

Teachers in conventional classrooms often. ask learners to express and show their thought processes by speaking out loud, drawing on a whiteboard, or even using physical objects.

Email contact: t-caca@microsoft.com



amazonmechanical turk
Artificial Artificial Intelligence
Your Account
HITs
Qualifications
201,223 HITs available now
Sign In
All HITs | HITs Available To You | HITs Assigned To You
Find HITs containing that pay at least $ 0.00
for which you are qualified
require Master Qualification
GO

Timer: 00:00:00 of 2 hours

Want to work on this HIT? Query verzenden

Total Earned: Unavailable
Total HITs Submitted: 0

Make Outbound Telephone Call To Verify Business Information
Requester: GTR_Co — Reward: $1.75 per HIT — Duration: 2 hours
Qualifications Required: Access and Availability Survey is 100

These are tests so the pay will eventually change once the kinks get worked out and we will issue qualifications for this. Once we do that we will make the instructions shorter since people will already know much of this information.

You or someone you are calling about has recently joined the MEDICAID plan. Please DO NOT say MEDICARE. It is completely different and that will make a mess. medi...CAID. It might be easiest if you say you are calling about someone else but up to you.

PLEASE be aware that the healthplan listed offers Medicaid. If they just say yes we take Medicaid that is NOT enough. Do they take that healthplan's Medicaid?

Calling the phone number listed in the directory. If asked how you got it please say the provider directory for the below company.

You can say you are doing some research for this person so just want to confirm the information and see about appointment availability. Hang up after 10 minutes if no one has answered call.

You want to confirm if the doctor listed below is in this health plan and how soon you would be able to get an appointment with that doctor. You also need to confirm the address and specialty listed here. You can use your own name and info if want does not really matter but may be easier. These calls are not secretive.

The provider may have other addresses please do not put those into comments unless part of question. If THIS address is valid for visits then put that if it is not answer accordingly.

Do not answer questions that you could not. So if the phone number is incorrect just answer that and submit. Do not answer No for the other questions. Just leave them blank.

You will not actually make an appointment just getting the info. You do not have to ask any questions verbatim, just be yourself to get the information.

*Company: UnitedHealthcare Community Plan of NY (Medicaid)
*Doctor Name: John Ioia
*INITIAL PHONE: (866) 274-9067.
*SECONDARY PHONE: 845-339-6122
*State Person Lives In For Scenario: NY.
*Address For The Doctor: 367 BROADWAY Kingston NY

*Special Instructions: You should say you are calling about a joint because this doctor is a Orthopaedic Surgeon. We know they will need referrals for specialists. Please see if you can still find out how early an appt might be available.

*You are not selling anything. These is a legitimate review of a health care company to confirm demographic information. The work is very important so we do try to pay fairly BUT will also quickly block someone if they submit junk. Its rare but unfortunately does happen.

* You will probably be speaking with an office assistant which is fine. If they transfer you that is also fine. If they tell you to call a different number you will not do that and answer the questions accordingly.

*It is also very important that you ask about the SPECIFIC healthcare provider listed. Most are doctors but even if not that is fine but just ask about that person. If the staff tries to give you information on someone else no or you will have to see and call back later.

PLEASE be aware that the plan below IS a Medicaid plan. If they say Yes we take Medicaid that is not enough. Do they take THAT health plans Medicaid?

*You will be making an outbound telephone call to get the information. All of the calls are recorded and we will listen to the recordings before approving the HIT.

*You MUST use the values from the top of the HIT. If you do multiple of these please CHECK because they will be different. Sometimes you might see the same doctors name for a different HIT but the address or phone or specialty is likely different.

*In most cases they will not offer to call you back if busy. If they do it is up to you BUT you would then need to call them back anyway. You MUST initiate the call though so if you play phone tag and they call you, then you need to come up with an excuse and call them right back on the recorded line.

*Do not offer to send us the recording from your answering machine or something, we cannot accept that.

*We do not have the office hours and are assuming most of these are 9 to 5. Some offices are closed certain days of the week though like Wednesday. If you call and the office is closed then just answer accordingly for the HIT. Those are approved but not bonused. We will resubmit another time.

*The scenario is pretty general on purpose. You are calling to get more information so treat it like a general inquiry call which is what it is.

**Comments: There are 2 comments areas availabile. One dealing with the information you received and one for HIT comments. You do not have to comment at all. But if you do:

Information Comments - Most of the time the information comments are used to capture updated information if provided. If for some reason the answers do not fully capture everything then please comment. PLEASE PLEASE do not put general or customer service comments here like the person was very nice or the opposite. We do not have time to read those nor will the company.

HIT Comments - If there is something you need to tell us about the HIT then put those here. These are new so we might not have everything worked out.

How To Complete HIT

1. Dial the INITIAL NUMBER at the top of this HIT. The system will ask you if you are ready for the "Angel ID".

2. The IVR system will read a 6 digit code. You must enter this code into one of the questions. The system will tell you to enter "9" if you need to repeat it.

3. Now the IVR system will tell you to enter the 10 digit number to call. You can either say, or enter the SECONDARY NUMBER listed above. PLEASE review this if you do multiple of these HITS because it CHANGES.

4. Complete the scenario at the top of this HIT. You DO NOT have to ask the question verbatim. Try to sound normal but you must ask the information in the scenario.

Pay SPECIAL attention to the type of insurance. Medicaid and Medicare are completely different.

5. The system will then connect your call like any other. Remember you must have followed the first step and dialed the special number so that the call is also recorded.

6. At end of call answer questions. Please ONLY use comment box to provide summary facts. Dont use to say "great job" or "answered correctly" etc.

Enter 6 digit Angel ID read at start of call. This is only way call is recorded.

Did the telephone number connect with a location even if incorrect?

- Yes
- No Answer
- Wrong Number
- Constant Busy Signal
- Answering Machine
- Out of Order Message
- Business Closed
- On Hold 10 Minutes or More

Was the telephone number correct for this provider at this location?

- Yes
- No

Does this provider work at this location?

- Yes
- No

Was this providers specialty correct for this location?

- Yes
- No

Does This provider accept the plan above?

- Yes
- No

Is this provider accepting new patients

- Yes
- No

What is earliest date

Please Enter The Name of The Person You Spoke To (Spelling Not Important)

Please Provide Any Overall Comments on Responses If Needed

Please Provide Any Overall Comments on HIT If Needed

Want to work on this HIT? Query verzenden

FAQ | Contact Us | Careers at Mechanical Turk | Developers | Press | Policies | Blog | Service Health Dashboard
©2005-2015 Amazon.com, Inc. or its Affiliates
An amazon.com. company

Figure 5.6: Comparison Between Different Amounts of Visual Areas
The upper HIT contains a small amount of visual areas,
while the lower HIT contains a very large amount of visual areas.

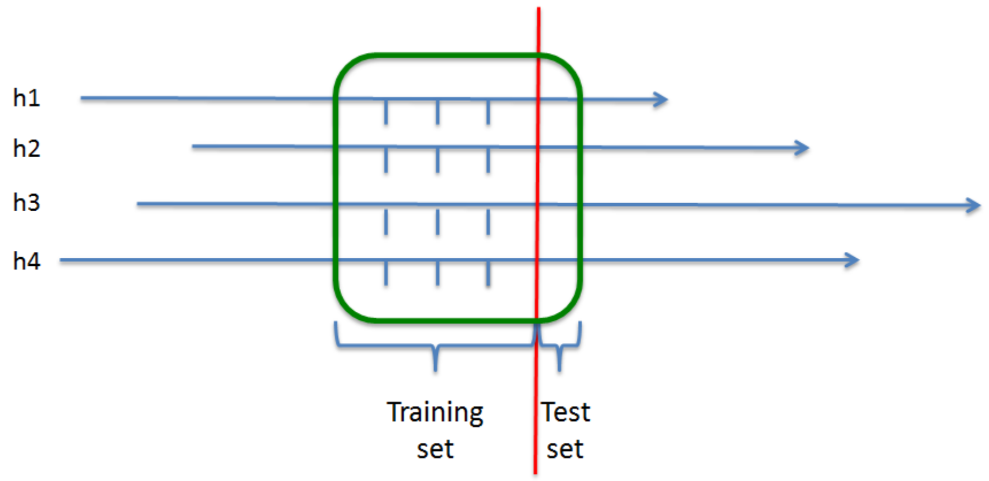


Figure 5.7: Throughput Prediction

which we also used. The training set thus consists of the market data for the four hours preceding time T. A single data entry for the algorithm has several dimensions:

- **Market Data:** The market data for the given hour: amounts of HITs and rewards posted and completed on MTurk. These attributes differ per hour.
- **HIT Metadata Attributes:** Metadata attributes *A1* through *A9* for the HIT. These attributes do not change over time.
- **HIT Content Attributes:** Content attributes *A11* through *A21* for the HIT. These attributes do not change over time.
- **HIT Throughput:** The throughput (amount of HITs completed) of the given HITgroup in the given hour. This value differs per hour.

Because the dataset by Difallah et al. was not collected using our extended tracker, we could not gather visual features, and thus they were only included in the part of the experiment using our own data. We will refer to the dataset we collected ourselves as Dataset A, and the dataset used by Difallah et al. as Dataset B.

5.3.2 Results - Dataset A

Table 5.3 shows the mean absolute error from various prediction models. A lower number indicates a better performance out of the model: it means that generally, its predictions were closer to the actual value. We used the following prediction models: Linear Regression, Ridge Regression, Lasso and Random Forest. Linear Regression performed horribly, Ridge Regression and the Lasso method performed reasonably well, and Random Forest performed the best by far out of all four. A graphical representation of the predicted versus actual throughput of the Random Forest model is shown in Figure 5.8.

Model	Mean Absolute Error
Linear Regression	3.43371548565e+13
Ridge Regression	13.4852059674
Lasso	13.365243909
Random Forest	**1.47431034483**

Table 5.3: Throughput Prediction Performance - Dataset A
Mean Absolute Error for various prediction models.
The mean throughput value was 38.8275862069 ± 101.612624442.

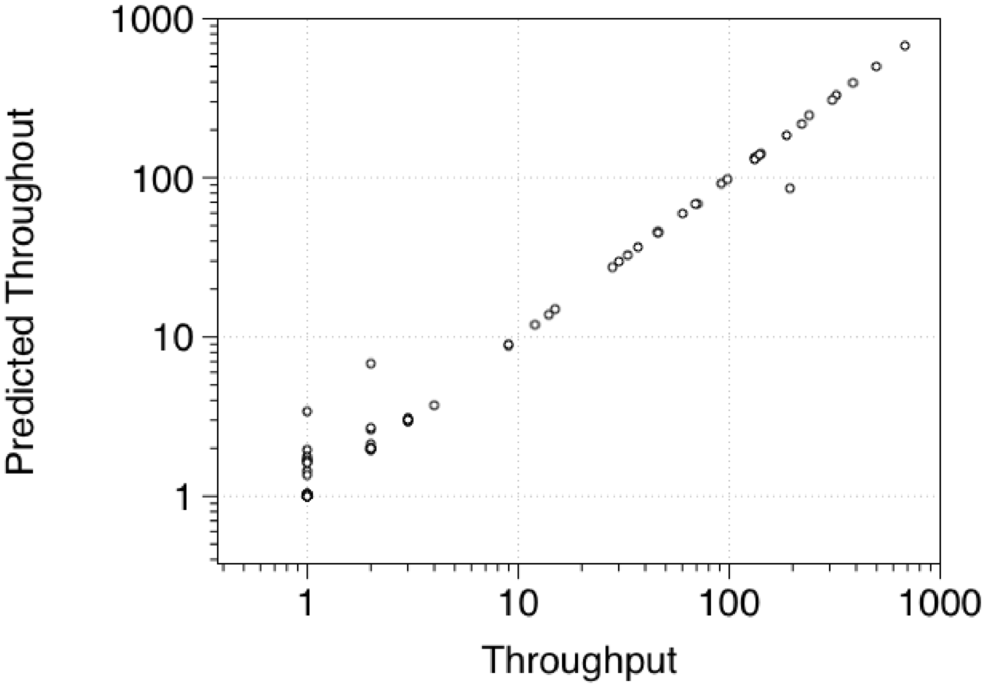


Figure 5.8: Throughput Prediction Results - Dataset A
Predicted versus actual throughput using a Random Forest model.

Based on these results, we conclude the Random Forest model performs very well in predicting throughput, and that just like in the experiment by Difallah et al., the prediction works better for larger batches. Table 5.4 shows the contribution scores for the top 25 attributes that contributed most to the prediction, sorted by absolute score. A positive score indicates a positive correlation between the attribute and throughput, and a negative score indicates a negative correlation.

We make several observations based on these results:

- **Market movement has the most influence.** Attributes from market data are the most important indicators for task throughput. The amount of initial hits is by far the most significant attribute, just as it was for Difallah et al. Additionally, statistics that indicate how many HITs are available on the market (*hitsCompleted, rewardsArrived, hitsAvailableUI*) correlate positively with throughput,

Attribute	**Contribution Score**
initialHits	59.48951914
unigram_shot	29.74892506
totalApproved	24.07984961
unigram_content	14.16681725
hitsCompleted	12.69597995
rewardsArrived	10.30126474
unigram_chinese	9.266171157
rewardsCompleted	-4.746358056
unigram_blocked	4.176982292
unigram_htm	3.039773852
unigram_ny	-2.996503414
unigram_violence	-2.661005612
hitsAvailableUI	2.591188684
unigram_case	2.455054234
unigram_click	-2.429056986
unigram_related	-2.18584631
unigram_reflects	1.833589004
hitGroupsArrived	1.638162625
unigram_play	-1.402751178
unigram_http	1.359896428
approvalRate	1.324680862
unigram_requires	-1.269983169
unigram_easy	-1.219236677
unigram_64	-1.206663617
hitGroupsCompleted	1.061824622

Table 5.4: Top Contributors to Prediction Score - Dataset A

while *rewardsCompleted* shows a negative correlation: when more rewards are being handed out, the overall market size will decrease and, accordingly, so will the availability of labour.

- **Semantic features improve prediction quality.** Though we do not see content or visual features among the top contributors, semantic features, specifically unigrams, have a big impact. For some, the relationship between the unigram and the task type is clear, such as *unigram_chinese*. Also interesting are the negative correlations: workers are apparently less interested in HITs that are presented as easy (*unigram_easy*), that might contain offensive content (*unigram_violence*) or that are presented as a game (*unigram_play*).

- **Qualifications attract workers.** The *totalApproved* attribute, along with the *approvalRate* attribute it correlates with, is also an indicator of high throughput. We expect that workers who qualify for common qualifications (such as needing a certain approval rate) will prefer to do tasks requiring these qualifications, as they tend to pay better.

5.3.3 Results - Dataset B

The throughput prediction experiment was also performed on the dataset used by Difallah et al. This dataset consists of MTurk market data collected from 2009 through 2014. This dataset contains metadata for all tasks and market statistics, and (slightly truncated) HTML content for slightly over half of the tasks, but not all. The dataset Difallah et al. used for throughput prediction in their own paper, which we also use, was a subset of this 5-year dataset. It consisted of data collected from June to October 2014. This is approximately a year before our own measurements, and in our other observations we already confirmed that data such as market statistics and task type distribution in our own data corresponded with observations in recent work by (among others) Difallah et al. Thus, as the two datasets represent similar information from the same market and no significant (for our purposes) changes occurred on the market between 2014 and 2015, we believe the two datasets are similar enough that we can compare them in this experiment. In other words: we can make the assumption that both datasets came from the same market, and were collected under the same market conditions.

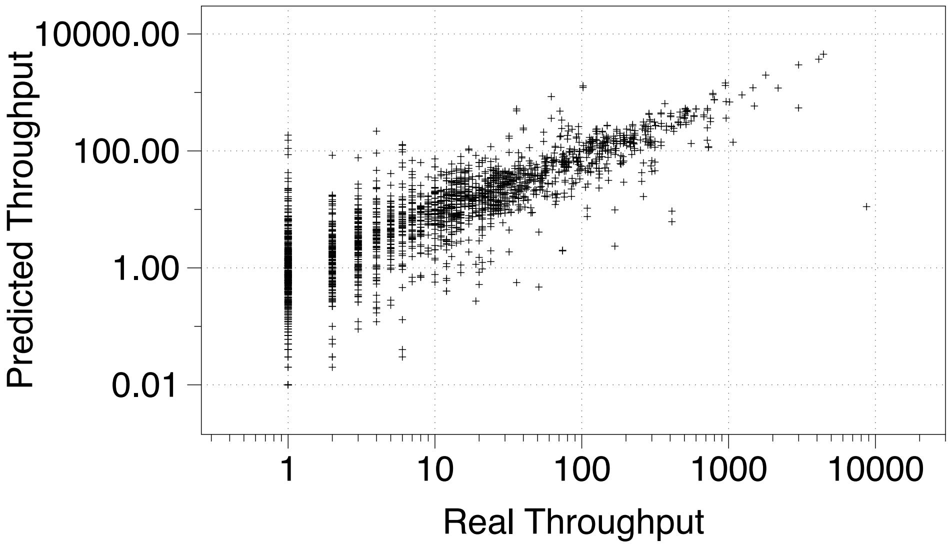


Figure 5.9: Throughput Prediction Results - Dataset B
Predicted versus actual throughput of the 5-month dataset, using metadata and semantic features.

Table 5.5 and Figure 5.9 show the results of our throughput prediction experiment on the five month dataset. Like Difallah et al., we used 50 evenly spaced time points in this interval, generating training and test data as described above. An important difference in our method for selecting data points was that we adhered to our goal of maximizing task diversity from the previous experiment. For every time point, we only used at most one HITgroup *per task type per requester*. This was made possible by the fact that the dataset was annotated with task type by Difallah et al. An example: if a requester posted two surveys in a given hour, only one would be incorporated in the

Feature Set	**Regression Model**		
	Linear	Lasso	Random Forest
Metadata	18.35	18.11	**17.45**
Dynamic	20.31	18.95	**20.47**
Semantic	576.44	**16.86**	20.53
Semantic LDA	18.92	**17.88**	19.75

Table 5.5: Throughput Prediction Performance - Dataset B
Mean Absolute Error of throughput prediction using different feature sets and regression models. The mean throughput value was 33.93 ± 67.17.

experiment, but if they posted two survey and two Content Access tasks, one HITgroup of each category would be included for a total of two tasks by that requester. We believe this significantly reduces the effect dominant requesters have on the experiment, and will come back to this when we discuss the results. In order to better examine the contribution of various types of attributes to prediction, we split up the prediction into an analysis of four different attribute groups. These were the following:

- **Metadata:** metadata features such as reward and time allotted.
- **Dynamic:** market dynamics features such as rewards available or amounts of HITs completed in the past hour.
- **Semantic:** semantic features such as keywords and unigrams in the text.
- **Semantic LDA:** semantic features modeled as a set of topics by Latent Dirichlet allocation (LDA) [4]. In other words, a method of reducing the very large amount of single semantic features to a smaller set of 'combined' features.

We observe that the mean absolute error for the experiment on the large dataset is relatively large compared to the average throughput value. Unfortunately, because Difallah et al. did not report their own results, we cannot directly compare our own results to theirs. We do note that their observation that the prediction works better for larger batches holds in our case, as well - this can be clearly seen in Figure 5.9. We also note that the mean absolute error values for each of the four considered feature sets are not very different. The experiment performed by Difallah et al. considered metadata and dynamic attributes, while our own experiment added the semantic and semantic LDA categories, which perform at the same level as the other two categories. We believe this shows the potential of using semantic features in throughput prediction, which we expected based on the experiment on Dataset A.

Table 5.6 shows the top 25 task attributes by absolute contribution to throughput prediction, corresponding to the prediction in Figure 5.9. These results are different both from the results we obtained for Dataset A, and the results obtained by Difallah et al. We see that the keywords 'tagging' and 'fast' are by far the most important features in this prediction. We believe this is due to a combination of these terms simply being popular among workers searching for new work. Furthermore, the popularity of the

Attribute	Contribution Score
keyword_tagging	108.4550737
keyword_fast	48.51691111
master qualification	21.62420036
keyword_ocr	-16.11195402
keyword_images	16.01048183
keyword_science	-15.29876699
location_us	-11.49538065
keyword_categorize	-8.211989903
keyword_categorization	6.75918132
keyword_data	5.490539861
keyword_mining	2.504987235
keyword_transcription	-2.102191942
no. of links	-1.730408269
title length	-1.629177075
no. of external links	1.404026731
keyword_object	1.295085654
keyword_instances	1.250865916
reward	-1.08954962
body text percentage	0.186764769
no. of images	0.099471103
keyword_objects	0.079962673
keyword_contour	0.061641002
no. of keywords	-0.02792923
HITs available	0.021000566

Table 5.6: Top Contributors to Prediction Score - Dataset B

’tagging’ keyword is likely due to the fact that a tagging task typically has a large batch size, as it usually requires workers to tag large amounts of items.

There are several other interesting insights that can be gleaned from these results. Firstly, this might be an insight into worker preferences: from this data, we might hypothesise that workers are not attracted to character recognition and transcription tasks (*keyword_ocr, keyword_transcription*), while they are interested in HITs dealing with image annotation (*keyword_images*). Of course, these results are only a first indication of possible preferences, and additional research would be required before any conclusions can be drawn.

Another observation is that keywords have replaced unigrams as being important semantic features compared to Dataset A. This makes sense, as unigrams will grow much more diverse as the amount of HITs studied increases, while the set of keywords can be expected to grow far less quickly, and a given keyword can be expected to occur more often in different HITs than a given unigram as the dataset increases in size. Other semantic features, such as body text percentage and the amounts of links and images on a page have also become more important compared to other features.

Metadata attributes also play a large role in the prediction: the master qualification seems to have a positive relationship with throughput, while restricting workers to only US-based ones decreases throughput. We already theorized that the master qualification seems to motivate and attract master workers effectively, and it is no surprise that reducing the amount of workers eligible to complete a task by half of all workers means that a task is not completed as quickly on average.

Finally, we observe that in this prediction, market statistics play a very small role. While our results for Dataset A and the experiment by Difallah et al. showed they played a very big role, the top 25 contributors for Dataset B only contain the HITs_available attribute. Several other attributes such as hitsCompleted, hitsArrived, diffHits and timeAllotted also contributed to the prediction, but only very slightly (absolute value <0.002). On the one hand, this could imply semantic features are effective at throughput prediction to the point where they play a bigger role than market dynamics. On the other hand, we did not measure attribute contribution in exactly the same way as Difallah et al. Their approach was to sample 1,000 test points and only use those points with an R-squared score >0, which included only 327 of the sampled points. For our own observation, we used all selected test points. It is possible that if we also only included test points with 'good' prediction scores, the HITs_available attribute would once again become more important. Additionally, as we noted before, we used an approach that maximized HITgroup diversity, rather than sample from the total amount of available tasks. Large requesters typically post batches of similar size, and these batches also make up the vast majority of the market. Workers also have previous experience with large requesters more frequently, which makes them either more or less likely to complete work by those requesters depending on previous experience or ratings on e.g. TurkOpticon. For these reasons, specific amounts of initial HITs could be indicators that a HITgroup was posted by a certain large requester, and thus has a certain expected throughput, which will have an increasingly stable value as the requester posts more HITs over time. Due to the domination of the market by a small amount of large requesters who post batches of similar size, a throughput prediction algorithm might perform very well on this vast majority of batches and thus use the amount of HITs available to detect them, while a batch of similar size posted by a different requester might not get the same results.

5.4 Subjective Task Complexity

Apart from approaching complexity as something that is objectively measurable, we also performed an initial experiment to measure the subjective complexity of a small set of reinstantiated HITs. Out of the 76 HITs we were able to visually reinstantiate, we were able to fully restore functionality for a set of 61. For those we were not able to restore completely, there were several reasons. In most cases, either we did not have enough original data (images to accurately reinstantiate the HIT fully, or the HIT contained functionality that we were not able to reproduce based on simply viewing the visually restored HIT.

These 61 reinstantiated tasks were posted on CrowdFlower along with a questionnaire intended to gather a complexity measurement. The questionnaire was based on the NASA-TLX (NASA Task Load Index) [24], which is *"a multi-dimensional rating procedure that provides an overall workload score based on a weighted average of ratings on six subscales: Mental Demands, Physical Demands, Temporal Demands, Own Performance, Effort and Frustration."* It has been applied, modified and evaluated in a variety of contexts since its inception in 1988, and was more recently also applied in contexts related to our own, such as information retrieval [58] and to assess subjective judgement difficulty for sentiment analysis tasks on MTurk [46].

The workers completed the HIT as follows:

1. Worker is shown an introduction page explaining they were to complete a task and answer questions about how they perceived the task's complexity.

2. Worker completes a reinstantiated task. We collect the data they input as part of the task as well as data on their behaviour by tracking mouse clicks and key strokes.

3. Worker completes the NASA-TLX questionnaire. First, the worker rates the task on each of the subscale categories mentioned above, and then weights the importance of each factor. This is done by presenting the worker with pairs of subscale categories (e.g. mental demand versus physical demand) and asking them which was more important to their experience of workload.

Each reinstantiated task was completed by 15 distinct workers. We instituted a spammer detection method by repeating 3 out of the 15 pairwise comparisons in the weighting part of the questionnaire at the end: lower-quality workers can be expected to give less consistent answers when questions are repeated, and workers that take a very long or short time to complete the survey were also considered less trustworthy. These two measures together resulted in 15.6% of responses being discarded due to incorrectly answered control questions, and 5.8% due to taking too long or too short, for a combined 21.4%.

The mean complexity score for the 61 tasks was 63.78 ± 11.56. The mean scores per complexity factor are shown in Figure 5.10, while the mean values for the weights assigned to each factor are shown in Figure 5.11. We observe that the Mental and Effort categories usually score and are weighted highly, the Performance category is weighted highly, and both score and weight are low for the Physical and Frustration categories. The scores for the Mental and Physical categories are in line with our expectations, as performing crowdsourcing tasks is primarily a mental task. The low average scores in the Performance and Frustration categories imply workers were generally satisfied with the task and their own performance.

Based on the obtained complexity scores for these 61 tasks, we performed a (Pearson's *r*) correlation analysis to attempt to find a correlation between our objective task attributes and the subjective complexity scores. For the non-semantic features, this did

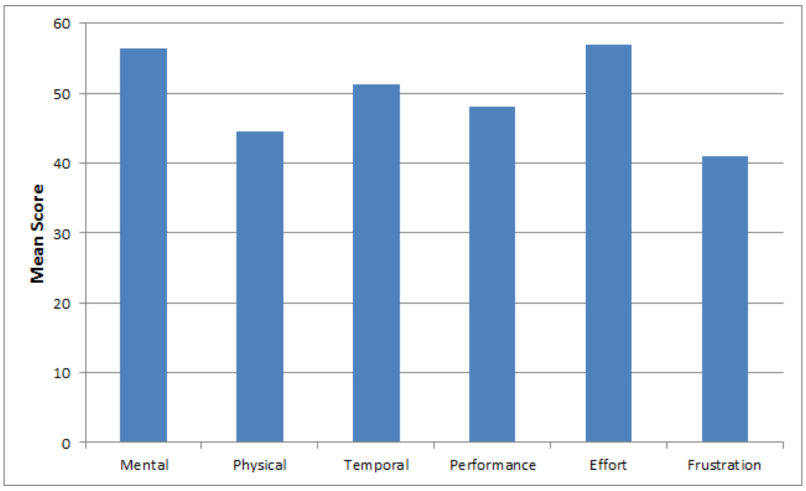


Figure 5.10: NASA-TLX Mean Scores
Mean scores assigned to each complexity category for the 61 reinstantiated tasks.

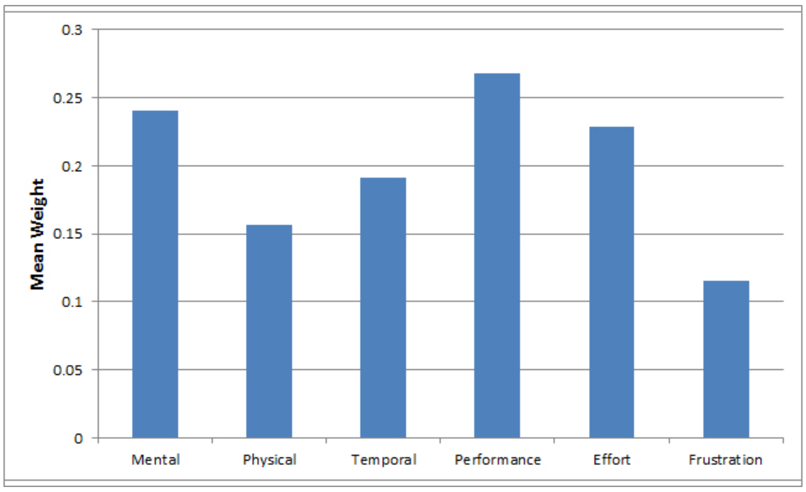


Figure 5.11: NASA-TLX Mean Weights
Mean weights assigned to each complexity category for the 61 reinstantiated tasks.

Attribute	***r***	**Attribute**	***r***
reward	0.27586	emphTextPct	-0.26907
totalApproved	0.13426	inputCount	-0.19136
wordCount	0.12330	initialHits	-0.17166

Table 5.7: Subjective Complexity Task Attribute Correlation

not allow us to find strong connections. The top 3 positive and negative correlations are shown in Table 5.7. We see a slightly positive correlation between subjective complexity and reward - it is to be expected that requesters will offer more money for tasks they consider to be more complex. We also see a slightly negative correlation with the percentage of emphasized text on a page: if a HIT page is easier to read because it is more structured, it will be easier to understand, but this will not have a dramatic effect on performing the task itself. None of the visual features showed any significant correlation with subjective complexity.

We also investigated correlations with keywords and unigrams, and found a small amount (16 unigrams, 7 keywords) that had a slightly stronger correlation with subjective complexity than any non-semantic attributes (absolute value of r between 0.28 and 0.36). However, because these are still not very strong correlations, and due to the fact that in a dataset of only 61 tasks there will only be a very small amount of measurements for a given unigram or keyword, we do not consider these results significant. If this experiment were to be repeated on a larger dataset, we expect performing this analysis again would deliver more reliable results, both for semantic and non-semantic attributes.

5.5 Task Design Guidelines

Based on the analysis of the data and the throughput prediction and subjective complexity experiments, we propose three task design guidelines for requesters. We consider tasks to be less complex when their throughput is higher, and will base our recommendations on the goal of presenting tasks that workers will not perceive as overly complex, with the goal of allowing them to complete those tasks more quickly.

1. **Watch the Market:** Apart from features inherent to a task itself, HIT throughput is highly dependent on the state of the market. If a lot of HITs are being posted, it will attract workers to the market, and depending on the time of day or day of the week, there will usually be more workers available to complete HITs. Planning the posting of HITs around busy times should improve the speed with which requesters can get results. Additionally, workers prefer to work on large batches of tasks, so requesters should post large amounts of HITs at once if possible to improve the throughput of their tasks. While we found in our experiment on Dataset B that statistics do not necessarily contribute to throughput prediction alongside other attributes, existing literature (by Difallah et al. [11], but also by Ipeirotis [30] and Kosara et al [40]) implies that market activity varies based on the time of day and day of the week, and that it is an important factor in task completion rate.

2. **Use MTurk Features to Your Advantage:** We observed that qualifications are not commonly used, but that they can help improve throughput by serving as a way to attract the most experienced or skilled workers to a task. We also noted that the title and description of HITs are typically used interchangeably, with the description repeating the title in most cases rather than including additional information. Some HITs use very few keywords or none at all, while our results

show that tasks with certain semantic features, such as certain keywords, are more likely to have higher throughput.

3. **Keep Your HIT Simple:** In our content analysis, we observed that there are values for the amounts of images, script files, word count and other content attributes that can be considered typical. We recommend that HIT designers examine the MTurk market and build their own HITs to resemble what workers are used to: the size, structure and content of a typical HIT should provide a good baseline on which to construct new ones. The subjective analysis also showed that tasks with a high word count or a large amount of links on the page tend to be experienced as more complex by workers.

Chapter 6

Conclusions and Future Work

In this chapter, we conclude the thesis by first summarizing and reflecting on our work, and finally looking forward. First, we summarize the most significant observations based on our results by linking them to our research questions. We reflect on our process and results, and finally single out directions for future work building on our own.

6.1 Conclusions

RQ1 *How can we measure the complexity of a human computation task?*

In Chapter 2, we described Campbell's framework for measuring complexity. We defined the complexity of a human computation task as follows:

The **complexity** of a HIT can be approximated by the sum of the complexity of its objectively measurable attributes, as perceived by its workers.

Based on this definition, we singled out three categories of task attributes that might contribute to task complexity. These were: **Metadata**, the attributes of a task relating to the way it is posted on a human computation platform; **Content Attributes**, attributes relating to the HTML content and semantic features of a HIT, and **Visual Attributes**, attributes relating to the visual complexity of a page. In order to measure these attributes, we constructed a real-time human computation market data tracker to gather market data from Amazon Mechanical Turk, and an attribute analysis pipeline to process this data and obtain measurements for task attributes. As an objective measure of complexity, we considered *task throughput*, the idea behind this being that since tasks that are less complex can be completed more quickly, batches of less complex tasks will have a higher throughput.

RQ2 *What are the effects of the complexity of a HIT on its throughput?*

We found that there are several categories of task attributes that affect throughput. These included market attributes, metadata attributes such as qualifications, and semantic attributes of task content. Non-semantic content features and visual complexity features did not seem to contribute significantly to the complexity of a HIT. The link between HIT complexity and throughput deserves further research, which we will discuss in the last section of this chapter.

RQ3 *Can we predict the complexity of a human computation task based on its attributes?*

Using task attributes and market statistics, it is possible to predict the throughput of a HIT group reasonably accurately, and we consider throughput to be an acceptable approximation of complexity. Looking at the attributes with the greatest negative correlation with throughput, we note that these are semantic features indicating the type of HIT to be completed. Workers tend to shy away from HITs that are presented as being easy and fun, either because they are looking for a more straightforward task or because they expect these words were used to make difficult tasks more attractive. Looking at the greatest correlation with subjective complexity, we observe that a higher reward correlates slightly with increased complexity, and a higher percentage of emphasized body text on the HIT page correlates slightly with decreased complexity.

RQ4 *How can the concept of task complexity be used to support human computation task design?*

Based on our analysis of the data from the tracker and the results of the throughput experiment, along with our personal observations of the various HITs stored by our tracker, we formulated three pieces of advice for human computation task designers. Firstly, the state of the **market** is an important contributor to how quickly tasks will be completed, and should thus be kept in mind when posting human computation tasks. Secondly, **platform features** such as the option to accompany human computation tasks with descriptions, keywords, and requiring worker qualifications should not be overlooked, as they are an important way of communicating the content of a task to possibly interested workers. Thirdly, HIT designers should attempt to **create simple tasks**: by looking at the way the various content elements of a typical HIT are put together, designers can create tasks that are more accessible to workers and thus easier to complete.

6.2 Discussion/Reflection

In this section, we will reflect on our process and results, identifying some key points worth reflecting on.

Firstly, we used task throughput as an objective approximation of task complexity. Given that there was some existing literature supporting the relationship between task complexity and completion time, and that we could not find existing work providing a different way to measure complexity objectively, we decided on the definition used in this thesis. While it allowed us to perform the throughput prediction experiment successfully, it would have been interesting to also include a subjective or direct measure of complexity to be able to draw more conclusions from the relationship between task attributes and complexity.

Secondly, our tracker could have used additional features. Though the extended tracker already tracks much more content, it does not store Flash content, audio, or video, to name some options. Additionally, the method of extracting data was rather basic, causing the tracker to be unable to store data that was not stored in a straightforward way. For example, several HITs we tracked used a JSON file to keep track of several image files to be evaluated by the worker. Since our tracker did not store this JSON file, it could not store the images used in the task either, as they were not directly linked to in the HTML file. Overall, the tracker does what it is supposed to do: it stores HIT content including stylesheets, scripts and images, which was our goal at the start. We believe that we have built it in such a way that it should not be difficult to extend it to track and store additional content types, though this may require more significant amounts of storage space for the additional content than the current implementation.

Finally, we used a relatively small dataset for most of our analysis. Due to the development time required to implement the tracker and analysis pipeline, and the costs and time involved in crawling data from Mechanical Turk, we only crawled one week of data. We had also underestimated the amount of HITs that would be reinstantiable before we started crawling: at the outset, we did not expect such a small percentage of stored data would actually be usable. While more data with a higher diversity of requesters would have undoubtedly improved our analysis, we feel that the dataset we constructed in the end was usable because of its high diversity in the types of HITs from different requesters that were represented.

6.3 Future Work

We have already alluded to possible directions for future work in the rest of this thesis. In this section, we identify two main opportunities for future research based on the work presented in this thesis: task reinstantiation and task attribute analysis.

Firstly, we believe **reinstantiation of HITs** made possible by the tracker we created offers many possibilities for research. By crawling market data in this way, researchers will be able to collect a diverse set of human computation tasks to be used for a variety of purposes. One direction we suggest is to use these restored HITs to measure the *subjective complexity* of tasks: simply ask workers to perform a reinstantiated task, and ask them about their opinions of the task. We already presented our experiment using a modified version of the NASA-TLX questionnaire in Chapter 5, but due to the time limitations of this project we were not able to investigate this in-depth. We believe that the initial results are promising enough to warrant future research, and we have already created a set of reinstantiated tasks that could be used to support this research. In the future, it may be possible to obtain a measure of subjective task complexity, which can then be compared to the attribute measurements generated by the task attribute analysis pipeline. We believe such a process could lead to insights that could help improve task design.

Secondly, we consider **task attribute analysis** to have more potential beyond the results presented in this thesis. As stated earlier, we used a diverse but small dataset for most of our analysis. Gathering more data and repeating the same analysis over a larger, more diverse set of HITs should lead to more insights on the relationships between various types of task attributes and task complexity and task throughput. In particular, we consider semantic and visual attributes to have potential for further research. We consider the semantic features relevant because they showed promise in our experiments, and the visual features because we expect there is more potential we did not tap in our own work. Our reason for this is that we believe the dataset we collected for this work did not lend itself to visual complexity analysis very well. A more refined implementation of HIT reinstantiation and larger datasets, combined with both objective and subjective measures for task complexity, might lead to important insights in human computation task design and possibly human computation in general. We express our hope that this work might be a first step in this direction.

Bibliography

[1] Mohammad Allahbakhsh, Boualem Benatallah, Aleksandar Ignjatovic, Hamid Reza Motahari-Nezhad, Elisa Bertino, and Schahram Dustdar. Quality control in crowdsourcing systems: Issues and directions. *IEEE Internet Computing*, (2):76–81, 2013.

[2] Iwan Barankay. Rankings and social tournaments: Evidence from a crowdsourcing experiment. In *Wharton School of Business, University of Pennsylvania Working Paper*. 2011.

[3] Debarshi Basak. Task recommendation in human computation. Master's thesis, TU Delft, the Netherlands, 2014.

[4] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[5] Michael Buhrmester, Tracy Kwang, and Samuel D Gosling. Amazon's mechanical turk a new source of inexpensive, yet high-quality, data? *Perspectives on psychological science*, 6(1):3–5, 2011.

[6] Michael Butkiewicz, Harsha V Madhyastha, and Vyas Sekar. Understanding website complexity: measurements, metrics, and implications. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 313–328. ACM, 2011.

[7] Donald J Campbell. Task complexity: A review and analysis. *Academy of management review*, 13(1):40–52, 1988.

[8] Donald J Campbell and Karl F Gingrich. The interactive effects of task complexity and participation on task performance: A field experiment. *Organizational Behavior and Human Decision Processes*, 38(2):162–180, 1986.

[9] Krista Casler, Lydia Bickel, and Elizabeth Hackett. Separate but equal? a comparison of participants and data gathered via amazon's mturk, social media, and face-to-face behavioral testing. *Computers in Human Behavior*, 29(6):2156–2160, 2013.

[10] Dana Chandler and Adam Kapelner. Breaking monotony with meaning: Motivation in crowdsourcing markets. *Journal of Economic Behavior & Organization*, 90:123–133, 2013.

[11] Djellel Eddine Difallah, Michele Catasta, Gianluca Demartini, Panagiotis G Ipeirotis, and Philippe Cudré-Mauroux. The dynamics of micro-task crowdsourcing: The case of amazon mturk. In *Proceedings of the 24th International Conference on World Wide Web*, pages 238–247. International World Wide Web Conferences Steering Committee, 2015.

[12] Djellel Eddine Difallah, Gianluca Demartini, and Philippe Cudré-Mauroux. Mechanical cheat: Spamming schemes and adversarial techniques on crowdsourcing platforms. In *CrowdSearch*, pages 26–30, 2012.

[13] Anhai Doan, Raghu Ramakrishnan, and Alon Y Halevy. Crowdsourcing systems on the world-wide web. *Communications of the ACM*, 54(4):86–96, 2011.

[14] Steven Dow, Anand Kulkarni, Scott Klemmer, and Björn Hartmann. Shepherding the crowd yields better work. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 1013–1022. ACM, 2012.

[15] Carsten Eickhoff. Crowd-powered experts: Helping surgeons interpret breast cancer images. In *Proceedings of the First International Workshop on Gamification for Information Retrieval*, pages 53–56. ACM, 2014.

[16] Carsten Eickhoff, Christopher G Harris, Arjen P de Vries, and Padmini Srinivasan. Quality through flow and immersion: gamifying crowdsourced relevance assessments. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 871–880. ACM, 2012.

[17] Enrique Estellés-Arolas and Fernando González-Ladrón-de Guevara. Towards an integrated crowdsourcing definition. *Journal of Information science*, 38(2):189–200, 2012.

[18] Siamak Faradani, Björn Hartmann, and Panagiotis G Ipeirotis. What's the right price? pricing tasks for finishing on time. *Human computation*, 11, 2011.

[19] Karën Fort, Gilles Adda, and K Bretonnel Cohen. Amazon mechanical turk: Gold mine or coal mine? *Computational Linguistics*, 37(2):413–420, 2011.

[20] Ujwal Gadiraju, Ricardo Kawase, and Stefan Dietze. A taxonomy of microtasks on the web. In *Proceedings of the 25th ACM conference on Hypertext and social media*, pages 218–223. ACM, 2014.

[21] Jorge Goncalves, Denzil Ferreira, Simo Hosio, Yong Liu, Jakob Rogstadius, Hannu Kukka, and Vassilis Kostakos. Crowdsourcing on the spot: altruistic use of public displays, feasibility, performance, and behaviours. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 753–762. ACM, 2013.

[22] Simon Harper, Caroline Jay, Eleni Michailidou, and Huangmao Quan. Analysing the visual complexity of web pages using document structure. *Behaviour & Information Technology*, 32(5):491–502, 2013.

[23] Simon Harper, Eleni Michailidou, and Robert Stevens. Toward a definition of visual complexity as an implicit measure of cognitive load. *ACM Transactions on Applied Perception (TAP)*, 6(2):10, 2009.

[24] Sandra G Hart and Lowell E Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. *Advances in psychology*, 52:139–183, 1988.

[25] David Hasler and Sabine E Suesstrunk. Measuring colorfulness in natural images. In *Electronic Imaging 2003*, pages 87–95. International Society for Optics and Photonics, 2003.

[26] Matthias Hirth, Tobias Hoßfeld, and Phuoc Tran-Gia. Anatomy of a crowdsourcing platform-using the example of microworkers. com. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2011 Fifth International Conference on*, pages 322–329. IEEE, 2011.

[27] Mahmood Hosseini, Keith Phalp, James Taylor, and Raian Ali. The four pillars of crowdsourcing: A reference model. In *Research Challenges in Information Science (RCIS), 2014 IEEE Eighth International Conference on*, pages 1–12. IEEE, 2014.

[28] Jeff Howe. The rise of crowdsourcing. *Wired magazine*, 14(6):1–4, 2006.

[29] Shih-Wen Huang and Wai-Tat 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*, pages 621–630. ACM, 2013.

[30] Panagiotis G Ipeirotis. Analyzing the amazon mechanical turk marketplace. *XRDS: Crossroads, The ACM Magazine for Students*, 17(2):16–21, 2010.

[31] Panagiotis G Ipeirotis. Demographics of mechanical turk. 2010.

[32] Panagiotis G Ipeirotis and Evgeniy Gabrilovich. Quizz: Targeted crowdsourcing with a billion (potential) users. In *Proceedings of the 23rd international conference on World wide web*, pages 143–154. International World Wide Web Conferences Steering Committee, 2014.

[33] Panagiotis G Ipeirotis, Foster Provost, and Jing Wang. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 64–67. ACM, 2010.

[34] Lilly C Irani and M Silberman. Turkopticon: Interrupting worker invisibility in amazon mechanical turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 611–620. ACM, 2013.

[35] Melody Y Ivory, Rashmi R Sinha, and Marti A Hearst. Preliminary findings on quantitative measures for distinguishing highly rated information-centric web pages. In *Proceedings of the Sixth Conference on Human Factors & the Web*, 2000.

[36] Melody Y Ivory, Rashmi R Sinha, and Marti A Hearst. Empirically validated web page design metrics. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 53–60. ACM, 2001.

[37] Nicolas Kaufmann, Thimo Schulze, and Daniel Veit. More than fun and money. worker motivation in crowdsourcing-a study on mechanical turk. In *AMCIS*, volume 11, pages 1–11, 2011.

[38] Aniket Kittur, Ed H Chi, and Bongwon Suh. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 453–456. ACM, 2008.

[39] Aniket Kittur, Jeffrey V Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton. The future of crowd work. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 1301–1318. ACM, 2013.

[40] Robert Kosara and Caroline Ziemkiewicz. Do mechanical turks dream of square pie charts? In *Proceedings of the 3rd BELIV'10 Workshop: BEyond time and errors: novel evaLuation methods for Information Visualization*, pages 63–70. ACM, 2010.

[41] Edith Law and Luis von Ahn. Human computation. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 5(3):1–121, 2011.

[42] Tak Yeon Lee, Casey Dugan, Werner Geyer, Tristan Ratchford, Jamie C Rasmussen, N Sadat Shami, and Stela Lupushor. Experiments on motivational feedback for crowdsourced workers. In *ICWSM*, 2013.

[43] Greg Little, Lydia B Chilton, Max Goldman, and Robert C Miller. Turkit: tools for iterative tasks on mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 29–30. ACM, 2009.

[44] Andrew Mao, Ece Kamar, Yiling Chen, Eric Horvitz, Megan E Schwamb, Chris J Lintott, and Arfon M Smith. Volunteering versus work for pay: Incentives and tradeoffs in crowdsourcing. In *First AAAI Conference on Human Computation and Crowdsourcing*, 2013.

[45] Winter Mason and Duncan J Watts. Financial incentives and the performance of crowds. *ACM SigKDD Explorations Newsletter*, 11(2):100–108, 2010.

[46] Tanushree Mitra, CJ Hutto, and Eric 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*, pages 1345–1354. ACM, 2015.

[47] Gabriel Mugar, Carsten Østerlund, Katie DeVries Hassman, Kevin Crowston, and Corey Brian Jackson. Planet hunters and seafloor explorers: legitimate peripheral participation through practice proxies in online citizen science. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pages 109–119. ACM, 2014.

[48] Sucheta Nadkarni and Reetika Gupta. A task-based model of perceived website complexity. *Mis Quarterly*, pages 501–524, 2007.

[49] Gabriele Paolacci, Jesse Chandler, and Panagiotis G Ipeirotis. Running experiments on amazon mechanical turk. *Judgment and Decision making*, 5(5):411–419, 2010.

[50] Alexander J Quinn and Benjamin B Bederson. Human computation: a survey and taxonomy of a growing field. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1403–1412. ACM, 2011.

[51] Sujith Ravi, Bo Pang, Vibhor Rastogi, and Ravi Kumar. Great question! question quality in community q&a. In *Eighth International AAAI Conference on Weblogs and Social Media*, 2014.

[52] Katharina Reinecke, Tom Yeh, Luke Miratrix, Rahmatri Mardiko, Yuechen Zhao, Jenny Liu, and Krzysztof Z Gajos. Predicting users' first impressions of website aesthetics with a quantification of perceived visual complexity and colorfulness. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2049–2058. ACM, 2013.

[53] Jakob Rogstadius, Vassilis Kostakos, Aniket Kittur, Boris Smus, Jim Laredo, and Maja Vukovic. An assessment of intrinsic and extrinsic motivation on task performance in crowdsourcing markets. In *ICWSM*, 2011.

[54] Ruth Rosenholtz, Yuanzhen Li, and Lisa Nakano. Measuring visual clutter. *Journal of vision*, 7(2):17, 2007.

[55] Joel Ross, Lilly Irani, M Silberman, Andrew Zaldivar, and Bill Tomlinson. Who are the crowdworkers?: shifting demographics in mechanical turk. In *CHI'10 Extended Abstracts on Human Factors in Computing Systems*, pages 2863–2872. ACM, 2010.

[56] Aaron D Shaw, John J Horton, and Daniel L Chen. Designing incentives for inexpert human raters. In *Proceedings of the ACM 2011 conference on Computer supported cooperative work*, pages 275–284. ACM, 2011.

[57] Anthony Tomasic, John Zimmerman, Aaron Steinfeld, and Yun Huang. Motivating contribution in a participatory sensing system via quid-pro-quo. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pages 979–988. ACM, 2014.

[58] Robert Villa and Martin Halvey. Is relevance hard work?: evaluating the effort of making relevant assessments. In *Proceedings of the 36th international ACM*

SIGIR conference on Research and development in information retrieval, pages 765–768. ACM, 2013.

[59] Luis von Ahn. Human computation. 2005.

[60] Luis Von Ahn and Laura Dabbish. Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 319–326. ACM, 2004.

[61] Luis Von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, and Manuel Blum. recaptcha: Human-based character recognition via web security measures. *Science*, 321(5895):1465–1468, 2008.

[62] Jie Yang, Claudia Hauff, Alessandro Bozzon, and Geert-Jan Houben. Asking the right question in collaborative q&a systems. In *Proceedings of the 25th ACM conference on Hypertext and social media*, pages 179–189. ACM, 2014.

[63] SN Yendrikhovskij, FJJ Blommaert, and H De Ridder. Optimizing color reproduction of natural images. In *Color and Imaging Conference*, volume 1998, pages 140–145. Society for Imaging Science and Technology, 1998.

[64] Lixiu Yu, Paul André, Aniket Kittur, and Robert Kraut. A comparison of social, learning, and financial strategies on crowd engagement and output quality. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pages 967–978. ACM, 2014.

[65] Man-Ching Yuen, Irwin King, and Kwong-Sak Leung. A survey of crowdsourcing systems. In *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*, pages 766–773. IEEE, 2011.

[66] Xianjun Sam Zheng, Ishani Chakraborty, James Jeng-Weei Lin, and Robert Rauschenberger. Correlating low-level image statistics with users-rapid aesthetic and affective judgments of web pages. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1–10. ACM, 2009.

Appendix A

Glossary

This appendix provides an overview of frequently used terms and abbreviations.

Crowdsourcing: the act in which the crowdsourcing party solicits, through an open online call, the completion of a certain task from a crowd of individual workers in exchange for a reward.

Crowdsourcing System / Platform: an online system that facilitates the operation and administration of the crowdsourcing process.

HIT: Human Intelligence Task. A task posted on MTurk by requesters to be performed by workers.

Human Computation: the utilization by a computation system or process of human processing power to perform its function.

Human Computation System: an intelligent system that organizes humans to carry out the process of computation.

MTurk: Amazon Mechanical Turk, an online crowdsourcing platform.

Requester: A party offering work on a crowdsourcing platform.

Spammer: A malicious worker who purposely submits incorrect or incomplete answers to earn more money.

Worker: A person working to complete tasks in a human computation system, usually in exchange for a reward.