Attribute Learning using Joint Human and Machine Computation

Edith L. M. Law

August 2012
CMU-ML-12-106

ML
MACHINE LEARNING
DEPARTMENT

Carnegie Mellon

Attribute Learning using Joint Human and Machine Computation

Edith L. M. Law

August 2012
CMU-ML-12-106

Machine Learning Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee

Luis von Ahn, CMU (co-chair)
Tom Mitchell, CMU (co-chair)
Jaime Carbonell, CMU
Eric Horvitz, Microsoft Research
Rob Miller, MIT

Submitted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy

This research was funded in part by Microsoft Research Fellowship, Fine Foundation under grant 083R17SM and National Science Foundation under grant IIS0968487. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

for my family

Abstract

This thesis is centered around the problem of attribute learning – using the joint effort of humans and machines to describe objects, e.g., determining that a piece of music is "soothing," that the bird in an image "has a red beak", or that Ernest Hemingway is an "Nobel Prize winning author." In this thesis, we present new methods for solving the attribute-learning problem using the joint effort of the crowd and machines via human computation games.

When creating a human computation system, typically two design objectives need to be simultaneously satisfied. The first objective is human-centric – the task prescribed by the system must be intuitive, appealing and easy to accomplish for human workers. The second objective is task-centric – the system must actually perform the task at hand. These two goals are often at odds with each other, especially in the casual game setting. This thesis shows that human computation games can accomplish both the human-centric and task-centric objectives, if we first design for humans, then devise machine learning algorithms to work around the limitations of human workers and complement their abilities in order to jointly accomplish the task of learning attributes. We demonstrate the effectiveness of our approach in three concrete problem settings: music tagging, bird image classification and noun phrase categorization.

Contributions of this thesis include a framework for attribute learning, two new game mechanisms, experiments showing the effectiveness of the hybrid human and machine computation approach for learning attributes in vocabulary-rich settings and under the constraints of knowledge limitations, as well as deployed games played by tens of thousands of people, generating large datasets for machine learning.

Thesis Committee:
Luis von Ahn (CMU) – co-chair
Tom Mitchell (CMU) – co-chair
Jaime Carbonell (CMU)
Eric Horvitz (Microsoft Research)
Rob Miller (MIT)

Keywords: Attribute Learning, Human Computation, Games with a Purpose, Machine Learning

Acknowledgments

When I came to Carnegie Mellon University in 2006, human computation as a research area was still in its infancy. Since then, this research area has grown tremendously and became a topic of great interests to a wide variety of research communities, including artificial intelligence, machine learning, human-computer interaction, data mining, computational economics and domain-specific disciplines such as computer vision, music information retrieval and natural language processing. I feel extremely privileged to have been in the position to witness the birth and growth of human computation, and to have the opportunity to actively shape this new research area from the beginning.

I am thankful for the support and mentorship of my co-advisors – Luis von Ahn and Tom Mitchell – whom I see as research pioneers in their respective disciplines. They gave me the complete freedom to create my own research agenda and take on ambitious projects, while remaining patient and supportive through challenging times. I have benefited greatly from their wisdom and broad (and sometimes unconventional) perspectives on what makes great research, and I will certainly always remember their kindness.

This thesis would not be possible without certain people who acted as my mentors and collaborators over the years – including Eric Horvitz (MSR), Rob Miller (MIT), Burr Settles (CMU), Max Chickering (MSR), Roger Dannenberg (CMU), Serge Belongie (UCSD), David Parkes (Harvard), Paul Bennett (MSR), Haoqi Zhang (Harvard), Jaime Carbonell (CMU), Yiling Chen (Harvard), Raman Chandrasekar, Maria Klawe (Harvey Mudd) and Miyoko Chu (Cornell), who have taken the time to provide detailed suggestions on my work, supported my effort to grow the human computation research community, and with whom I had worked on a variety of interesting research ideas.

Most importantly, I want to thank my parents, my husband Eric Blais, my friends Emily Stiehl, Brian Ziebart, Jenn Tam, Sean Hyde, Lucia Castellanos, and Polo Chau for their companionship, unconditional love and support. Last, but not least, to my beloved Jacob and baby no. 2 – thank you for your patience and resilience as your parents embark on their research careers; without you, my life would not have been nearly as rich and fulfilling.

To me, the Ph.D. has been much more than just a journey of learning – it was a journey that involved leaving a home that I love, growing a family, making new and endearing friendships, realizing my potentials and overcoming my limitations, and entering an amazing world where people from all over the globe can be readily connected through the shared passion for a single idea. This entire experience has been challenging, but extremely interesting and rewarding. I look forward, with great excitement, to the next stages of this journey.

Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Motivation

How do we recognize the objects that we see? What mood does a piece of music evoke? While the question of exactly how humans perceive is an important scientific one, our ability to build machines that can *mimic* human perception, i.e., identify objects and classify music, etc., can help address one of the most pressing technological problems today—the need to organize the billions of multimedia objects (images, music and videos) on the Web so that users can find the needles in the haystack with little effort.

Unfortunately, perception is a difficult problem for machines; and training machine learning algorithms to perceive like humans necessitates a huge amount of labeled data, which is typically costly and time consuming to collect. To address this challenge, human computation systems – systems that elicit the help of human workers to perform computation – were introduced. The ESP Game, for example, is a human computation system that maps images to tags, by engaging humans to play a game in which they are rewarded each time they agree on a description for an image. It was shown that these so-called *Games with a Purpose* are a reliable way to quickly collect millions of accurate descriptors for images, music, and videos, which can then used to index objects on the Web and facilitate search.

This thesis is centered around the problem of attribute learning – using the joint effort of human game players and machine learning algorithms to describe objects, e.g., to determine that a piece of music is "soothing", that the bird in an image "has a red beak", or that Ernest Hemingway is an "Nobel Prize winning author". Attributes are compoundable, making them extremely useful for information retrieval (e.g., complex queries such as "asian women with short hair, big eyes and high cheekbones") and identification (e.g., find an actor whose name you forgot, or an image that you have misplaced in a large collection). In recommendation systems, indexing objects by attributes make it possible to explain the reason why a particular item is chosen for the user (e.g., this song is recommended to you because it is "calm" and "sentimental" just like the other ones that you like). Consider the following concrete applications where attribute learning would be useful:

Scenario 1: Naming Plants
Alice saw a beautiful tree in the Redwood National Park. Being a novice botanist, she was really interested in knowing the identity of the species. Luckily, there is a tool that allows Alice to enter attributes such as "the tree is found on the West

Coast," "the leaves have saw-like edges, and are long and narrow," and retrieve a set of candidate species names (along with some representative images for each species).

Scenario 2: Identifying People
Bob is having a conversation with someone, debating who is going to win best supporting actress award in the upcoming Oscars ceremony. Bob remembers really enjoying the performance of this particular actor in a movie he saw, but could not quite remember her name. By answering a set of questions presented by the computer (e.g., "what movie did she star in?", "does she have brown hair?", "does she have thick eyebrows?", etc.), Bob is able to retrieve a set of candidate actresses along their headshots, and find the name of the actress he is looking for.

Scenario 3: Discovering Birds
Eve is doing a school projects about birds. She wants to find a set of birds that live in the tropics, that are "small", and have "brilliant colors". Eve is able to issue a complex search query and find a set of birds that fit those descriptions.

Scenario 4: Locating Objects
Co-bot is a robot that helps people, e.g., fetching objects that a person has misplaced. Imagine the following interaction: Tom is looking for his cup and asks Co-bot for help. Co-bot asks Tom a series of questions, e.g., "is a coffee mug?", "is it dark in color", "does it have writings on it?", etc. Equipped with the answers to these questions, Co-bot goes around the room and attempts to classify each object by those attributes, in order to locate Tom's cup.

1.2 Thesis Overview

This dissertation is centered around the problem of attribute learning – using the joint effort of human game players and machine learning algorithms to describe objects, e.g., to determine that a piece of music is "soothing," that the bird in an image "has a red beak," or that Ernest Hemingway is an "Nobel Prize winning author." In this thesis, we present new methods for solving the attribute learning problem via human computation games using the joint effort of the crowd and machines.

When creating a human computation system, there are typically two design objectives that need to be satisfied simultaneously. The first objective is *human-centric* – the task prescribed by the system must be intuitive, appealing and easy to accomplish. The second objective is *task-centric* – the system must accomplish the task at hand. These two goals are often at odds with each other, especially in the casual game setting; in designing a system that is human-friendly, the system often ends up falling short of accomplishing its tasks. This thesis shows that it is possible to design human computation games that accomplish both the human-centric and task-centric objectives, by first designing for humans, then devising machine learning algorithms to work around the limitations of human workers and complement their abilities in order to jointly accomplish the task of learning attributes. We demonstrate the effectiveness of our approach in two concrete problem settings.

In the first problem setting (chapter 3), we focus on the problem of learning attributes in a vocabulary-rich setting, where the input object (e.g., music) can be described using a vast and diverse set of attributes that may be synonymous or imprecise. We introduce a new game mechanism (and a deployed game called TagATune) that is capable of extracting music attributes. While TagATune is designed to be human friendly, the collected attributes are noisy, making them unamenable as training data for machine learning algorithms. We devise a new algorithm that is capable of leveraging the open vocabulary labels collected by TagATune to train music tagging algorithm, and evaluate its performance both in terms of data and time efficiency.

In the second problem setting (chapter 4), we address the problem of learning attributes under knowledge limitations, where either the attributes or entities are unfamiliar to the average worker. We present a new game mechanism called complementary-agreement (and an implemented game called The Perfect Split) for extracting attributes and attribute values, and evaluate its effectiveness in two contrasting case studies, namely bird image classification and noun phrase categorization. In the case of bird image classification, we show that instead of asking people to directly categorize bird images, a better approach is to collect perceptual attributes from humans, then use these human-generated attributes as features for training a classifier to predict the actual categories. In the case of noun phrase categorization, we show that transforming the representation of the entities (i.e., turning noun phrases into images) enables the crowd to verify categorical facts, despite the fact that most of noun phrases are obscure and unknown. Furthermore, we show that the complementary-agreement mechanism can be used to categorize noun phrases by images, showing that a game-based approach is feasible for engaging the crowd to supervise a continuous, never-ending web mining system like NELL (Never-Ending Language Learner).

1.2.1 Framework

In order to provide a unifying language for describing the attribute learning problem, games with a purpose and joint human and machine computation, we introduce here a framework for attribute learning, which we will refer to throughout this dissertation while addressing specific challenges.

Formally, the attribute learning problem can be described as the task of filling in a $N \times M$ matrix $\mathcal{V}$, where the rows are a set of entities e_i, $i = 1 \ldots N$, the columns are a set of attribute a_j, $j = 1 \ldots M$, and each cell of the matrix $v_{ij} = f_j(e_i)$ is the true value of the attribute a_j for entity e_i. For simplicity, we assume that $v_{ij} = [0, 1]$, representing the probability that the attribute describes an entity.

	a_1^*	a_2^*	a_3^*	.	.	.	a_M
e_1							
e_2							
e_3							
.							
.							
.							
e_N							

Here are a few examples of the $(e_i, a_j, v_{i,j})$ tuples in different domains:

- $e_i =$ national_anthem.mp3, $a_j =$ patriotic, $v_{ij} = 0.98$
- $e_i =$ specimen.jpg, $a_j =$ horns_is_long, $v_{ij} =$ 0.7
- $e_i =$ “Barack Obama”, $a_j =$ is_a_democrat, $v_{ij} = 1$
- $e_i =$ parade.mov, $a_j =$ people_dancing, $v_{ij} = 0$

One can see the matrix V as a huge matrix, which has a pre-defined set of entities and all the possible attributes in the world. The problem of attribute learning involves two subtasks – (i) *discovery*: determining which attributes (or columns of the matrix) are relevant to the entities and the task at hand, and (ii) *scoring*: extracting the true values of the attributes for each entity (i.e., fill in the cells of the selected columns of the matrix). In general, the relevant attributes are the ones that discriminate between different entities or different sets of entities. For example, if our objective is to categorize the bird images, then the relevant attributes (or columns of the matrix) are the properties of the birds (e.g., color and shape of particular body parts) that can be used to discriminate between different species. If our objective is to annotate music, then the relevant attributes are the ones that help to identify different pieces of music, allowing users to retrieve them with ease. In both cases, attributes that have values that are the same for all entities are not relevant or useful; for example, the attribute “has a head” will have the value 1 for every bird image, and the attribute “music” will have the value 1 for every piece of music.

The task of discovery and scoring can be accomplished by both machines and humans. There exist now web mining systems that can crawl the Web and collect attributes associated with a wide variety of entities. Alternatively, there are also human computation systems that collect attributes from paid crowdworkers and unpaid volunteers (e.g., gamers) about images, music, videos and named entities. Once the relevant attributes are collected, both machines and humans can infer the attribute values of an entity. A machine learner can, for each attribute a_j and entity e_i, learn a function $\hat{f}_j^m(r^m(e_i))$ that predicts the value of the attribute of the entity, where $r^m(e_i)$ is some machine-interpretable features that are typically extracted automatically (e.g, color histogram for images, spectrogram features of music). Humans can perform the same task $\hat{f}_j^h(r^h(e_i))$, where $\hat{f}_j^h$ is a human function that outputs the value of the attribute a_j given a human interpretable representation $r^h(e_i)$ of the entity.

In this thesis, we address two particular challenges that arise within this framework, specifically learning attributes when the entities can be described by a huge number of synonymous attributes, or when attribute values are difficult for humans to determine because of knowledge limitations. There are wide variety of attribute learning problems in this framework that are beyond the scope of this thesis, but the framework and the findings presented here should offer solid grounding for future work.

1.3 Summary of Contribution

At a high level, this thesis contributes new techniques for achieving the human-centric objective in human computation games, and demonstrates the learning challenges and opportunities these afford for the task-centric objectives, i.e.,

- new, generalizable game mechanisms for learning attributes for a variety of entity types, including music, images and text.
- new methods for combining humans and machines in a human computation system, in order to simultaneously achieve both the *human-centric* and *task-centric* objectives.
- a general framework and language for describing specific challenges associated with the attribute learning problem.

More concretely, this thesis addresses two types of challenges in Attribute Learning. First, using music tagging as a case study, we study the challenge of learning attributes for vocabulary-rich input data. Specifically, we introduce

- a new game mechanism called *input-agreement* that (i) poses an easier task for game players, and (ii) extracts more informative attributes than the widely adopted output-agreement mechanism.
- a deployed game called TagATune, which has interacted with tens of thousands of game players and collected over a million music tags, which in turn, became one of the largest content-based music tag dataset distributed to the music information retrieval research community. The TagATune game, as our experiments show, is useful not only for collecting music attributes, but also human evaluations of automated music tagging algorithms.
- a new machine learning algorithm that can be trained efficiently using crowd-generated, open vocabulary data collected by TagATune. This algorithm is (i) both data-efficient (i.e., can utilize an arbitrary open vocabulary of tags) and time-efficient (i.e., reduces training time by 94% compared to learning from tag labels directly), and (ii) achieves similar performance for annotation and

superior performance for retrieval when compared to the traditional approach of performing multiple binary classifications.

Second, using two case studies – bird image classification and noun phrase categorization – we investigate the challenge of knowledge limitations in attribute learning, where crowdworkers are asked to determine the attribute values when either the attributes or entities are unfamiliar to them. Specifically, we introduce

- a new game mechanism called *complementary-agreement* that can (i) extract attributes and attribute value explicitly, (ii) generate informative attributes that distinguish among sets of objects, and (iii) incentivize players to answer multiple-entities, binary choice questions truthfully.
- an implemented game called The Perfect Split (based on the complementary-agreement mechanism) and a user study evaluating the effectiveness of several modes of the game at collecting attributes and attribute values.
- experiments showing that a human computation system can achieve both the human-centric and task-centric objectives, by using machines to complement the abilities of crowd workers and to overcome their limitations. Specifically, for bird image classification, instead of asking crowd workers to directly categorize bird images, the system allows them to provide attributes of their own choosing, then trains a machine learning algorithms using this crowd-generated data to infer actual categories. In the case of noun phrase categorization, we showed that by transforming the representation of the entity (i.e., from noun phrases into images), we can overcome the knowledge limitations of crowd workers, enabling them to perform a task previously deemed impossible.

Chapter 2

Techniques for Attribute Learning: An Overview

There has been substantial research on attribute learning. In this section, we will review some of the related work that are most relevant to the challenges addressed by this thesis.

2.1 Human Computation

2.1.1 What is Human Computation?

Since 2005, the phrase "human computation"[1] [von Ahn 2008a] has become synonymous with many other equally loosely defined research areas, such as "crowdsourcing," "social computing," "socio-computational systems," and "collective intelligence." In clearly defining what we mean by "human computation," we can outline the scope of this research area, pinpoint the fundamental research questions, identify end goals and focus our efforts in reaching them.

To understand what we mean by "human computation," we must first define the word "computation." In our formulation, **computation** is the process of mapping of some input representation to some output representation using an explicit, finite set of instructions (i.e., an *algorithm*). In the classic work by Alan Turing, computation is similarly defined, where the input and output representations are symbols, the process is the writing of symbols in each cell of an unlimited tape, and the set of instructions or algorithm is a state transition table that determines what symbol should be written for the current cell. Similarly, a human computer who is given two quantities (input representation) and asked to multiply them together (explicit instruction) to generate a product (output representation) is performing computation. In fact, the Turing Machine was meant to mimic the capability of human computers in carrying out mathematical calculations. In Turing's own words, "the idea behind digital computers may be explained by saying that these machines are intended to carry out any operations which could be done by a human computer" [Turing 1950].

Following this definition, **human computation** is simply computation that is carried out by humans. Likewise, **human computation systems** can be defined as intelligent systems that organize humans to carry out the process of computation—whether it be performing the basic operations (or units of computation), taking

[1] Some of the materials in this section come from our previously published book on Human Computation [Law 2011a]

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). The meaning of "basic" varies, depending on the particular context and application. For example, the basic unit of computation in the calculation of a mathematical expression can be simple operations (such as additions, subtractions, multiplications and divisions) or composite operations that consist of several simple operations. On the other hand, for a crowd-driven image labeling system, a user who generated a tag that describes the given image can also be considered to have performed a "basic" unit of computation. In an experiment for solving the graph coloring problem using distributed human computation [Kearn 2006], the basic unit of computation each human solver performed was to "change the color of his or her node, given the current colors of the neighboring nodes." The system's job was to simply take the output from each human solver and update the state (i.e., color of each node) of the network.

By our definitions, human computation systems must involve humans playing a *conscious* role in determining the outcome of the computation. Therefore, volunteer computer projects (where people donate their idle CPU power to help solve large computational problems) or participatory sensing project (where participants are merely sensor carriers) are not considered human computation systems.

2.1.1.1 Explicit Control

How does the concept of human computation differ from other concepts, such as crowdsourcing, collective intelligence and social computing? In the spirit of crowd wisdom, let's first examine these concepts as they are defined in Wikipedia (Figure 2.1).

Table 2.1: Related Concepts [Wikipedia 2012]

Crowdsourcing	The act of outsourcing tasks, traditionally performed by an employee or contractor, to an undefined, large group of people or community (a crowd) through an open call.
Collective Intelligence	A shared or group intelligence that emerges from the collaboration and competition of many individuals and appears in consensus decision making in bacteria, animals, humans and computer networks.
Social Computing	Technology for supporting any sort of social behavior in or through computational systems, e.g., blogs, email, instant messaging, social network services, wikis and social bookmarking.
	Technology for supporting computations that are carried out by groups of people, e.g., collaborative filtering, online auctions, prediction markets, reputation systems, computational social choice, tagging and verification games.

Based on these definitions, “crowdsourcing” can be considered a method or a tool that human computation systems can use to distribute tasks through an open call. However, a human computation system does not need to use crowdsourcing; a system that assigns tasks to a closed set of workers hired through the traditional recruitment process (e.g., resumes, in-person interviews) can still be considered a human computation system.

The term “social computing” is a broad concept that covers everything to do with social behavior and computing. Human computation intersects social computing in that some, but not all, human computation systems require social behavior and interaction amongst a group of people. That is, human computation does not necessarily involve large crowds, and workers are not always required to interact with one another, either directly or indirectly (e.g., through a market mechanism).

Finally, “collective intelligence” refers to the emergent intelligent behavior of a group of individuals, which includes non-humans and non-living things. Collective intelligence, therefore, is an even broader concept that subsumes crowdsourcing, social computing and human computation. Consult [Doan 2011] for a detailed survey and alternative perspectives on Web-based mass collaboration systems.

Most importantly, none of the related concepts emphasize the idea of **explicit control**. In fact, they assume that a large part of the computational outcome is determined by the natural dynamics (e.g., coordination and competition) [Kittur 2008, Kittur 2007] between the individuals of a group, which the system cannot or does not deliberately control. There is no explicit decomposition or assignment of task, nor are there any explicitly designed mechanisms for ensuring that the human computers tell the truth. In reality, the amount of explicit control in crowd-driven systems is a continuum. For example, many crowd-driven systems (e.g., Wikipedia) do enforce rules, protocols and standards (such as “technical specifications” or “routinized processes” [Malone 1998]) by which individuals need to abide. What is different about human computation systems is the level of explicit control, which is on the greater end of the spectrum. In other words, instead of focusing on studying human behavior, the focus of human computation research is on *algorithms*, which either specify exactly what gets processed, by whom and how, *or* explicitly organize human efforts to solve the problem in a well-defined manner.

Conceptually, there are three aspects –“what,” “who” and “how” – of any human computation systems (depicted in Figure 2.1) where explicit control can be applied.

The “What” Aspect

In order to generate a solution to a computational problem, we must have an algorithm that outlines exactly how to solve the problem. An algorithm consists of a set of operations and a combination of control structures that specify how the operations are to be arranged and executed. Similar to algorithms in the traditional sense, some human computation algorithms are more efficient than others. For example, if our computational problem is to map a set of images to tags, an efficient algorithm would make use of machine intelligence (e.g., active learning [Settles 2012]) to se-

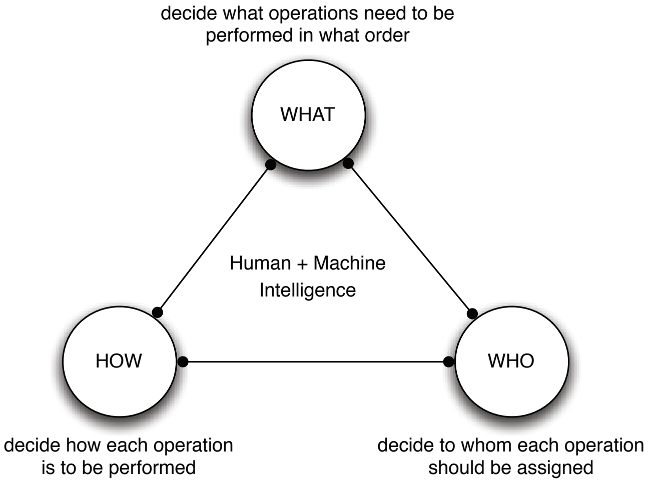

decide what operations need to be performed in what order
WHAT
Human + Machine Intelligence
HOW
WHO
decide how each operation is to be performed
decide to whom each operation should be assigned

Figure 2.1: Three central aspects of human computation systems.

lect only images that the computer vision algorithm does not already know how to classify. Such an algorithm can greatly reduce the costs of the computation, both in terms of time and monetary payment to human workers. Some research questions relevant to the "what" aspect of human computation include the following.

- What tasks can be performed adequately by machines, therefore eliminating the need for human involvement? Can we leverage the complementary abilities of both humans and machines [Horvitz 2007b] to make computation more accurate and efficient?

- How do we decompose complex tasks into manageable units of computation and order them in such a way to handle the idiosyncrasy of human workers?

- How do we aggregate noisy and complex outputs from multiple human computers in the absence of ground truth?

The "Who" Aspect

Knowing what operations need to be performed, the next question is to whom each operation should be assigned. While for some tasks, aggregating the work of non-experts suffices, other tasks are knowledge intensive and require special expertise. For example, a doctor who is asked to verify that the fact "Obacillus Bordetella Pertussis is a bacterium" is likely a better (and faster) judge than someone without any medical training. Some research questions relevant to the "who" aspect of human computation include:

- What are some effective algorithms and interfaces (e.g., search or visualization) for routing tasks?

- How do we model the expertise of workers, which may be changing over time?
- What are some optimal strategies for allocating tasks to workers, if their availability, expertise, interests, competence and intents are known versus unknown?

The "How" Aspect

Finally, the "how" aspect pertains to the question of design—how can the system motivate workers to participate and to carry out the computational tasks to their best abilities (i.e., truthfully, accurately and efficiently). Much of the work contained in this thesis addresses questions about this aspect of human computation. Some research questions relevant to the "how" aspect of human computation include:

- How do we motivate people to have a long-term interaction with the system, by creating an environment that meets their particular needs (e.g., to be entertained, to have a sense of accomplishment or to belong to a community)?
- How do we design game mechanisms [von Ahn 2008b] that incentivize workers to tell the truth, i.e., generate accurate outputs?
- What are some new markets, organizational structures or interaction models for defining how workers relate to each other (as opposed to working completely independently)?

2.1.1.2 Definition and Fundamental Questions

we adopt with the following condensed definition of human computation:

> "Given a **computational problem** from a **requester**, design a **solution** using both **automated computers** and **human computers**."

Following this definition, the five most fundamental questions in human computation are the following:

1. What computational problems can or should be solved using human computers?
2. What are different types of solutions to a computational problem? For more explicit solutions, what are some programming paradigms for designing algorithms with humans in the loop?
3. For a requester, how do we guarantee that the solution is accurate, efficient and economical?
4. For a worker, how do we design an environment that motivates their participation and leverage their unique expertise and interests?

5. How do we leverage the joint efforts of both automated and human computers as workers, trading off each of their particular strengths and weaknesses? What other roles does machine intelligence play in human computation?

2.1.1.3 Markets

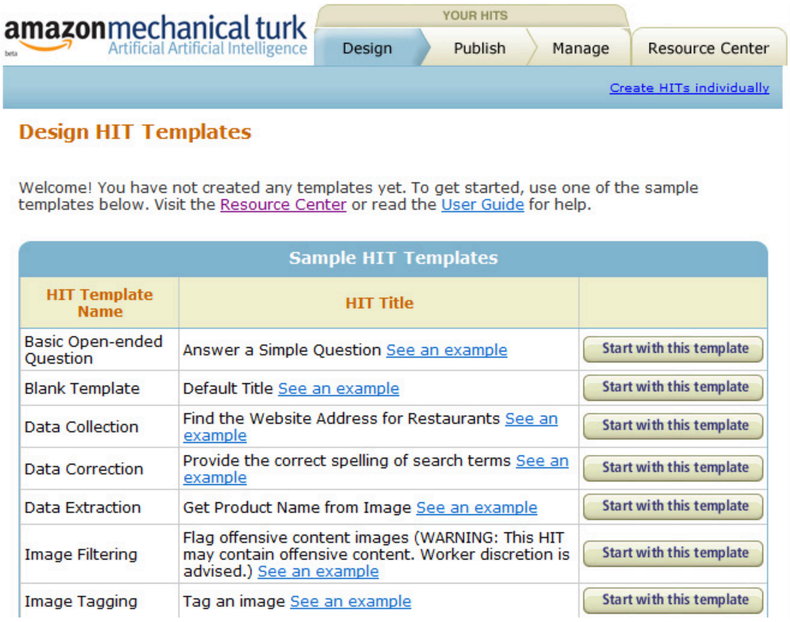
In human computation systems, a *market* refers to a pool of individuals who are available to work on the computation tasks at hand. The possible motivations for human computers to want to participate are varied, but there is one thing in common: the particular computational problem that workers decide to devote time and effort to help solve has significant value to them. This value is often more than just monetary. Workers might be seeking access to valuable resources, entertainment, the opportunity to contribute to the common good or learn something new; in return, they are willing to perform small units of computation. On the other hand, requesters are also the stakeholders of any human computation systems, whose goal is to solve the computational problem of interests in the most accurate, efficient and economical way possible. A human computation system is not sustainable without satisfying the needs and wants of both workers and requesters.

Usability is a concern in the design of any software systems; for human computation systems, it is no exception. First, a design that is easy to learn implies a low barrier of entry for new users and encourages them to revisit the system. This is especially important for human computation systems, such as games, where workers are volunteering their time and effort. If they do not find the task worth their while, it is not likely that they will continue to participate. In this section, we will focus on the two human computation markets – Mechanical Turk and Gamers – that are most relevant to this thesis. We will describe the characteristics of the workers in these two markets – who they are and what motivates their participation.

Mechanical Turk and Paid Crowdsourcing

There exist many different crowdsourcing platforms. The most well known is Amazon Mechanical Turk (AMT), released in late 2005 [MTurk 2012]. The name Mechanical Turk is borrowed from a 18th century machine called “The Turk,” a seemingly automatic chess-playing machine that is actually operated by a human in the background. Amazon Mechanical Turk provides a platform for *requesters* to post tasks to *workers* to perform in return for monetary payment. These tasks are called HITS, which stands for “Human Intelligence Tasks.” This service quickly grew in popularity. By 2010, there has been an estimated 400,000 workers on Mechanical Turk [Ross 2010].

Tasks distributed through Mechanical Turk are typically small (i.e., quick to complete), as are the monetary reward for each task—90% of the HITs have a reward of less than 10 cents [Ipeirotis 2010a]. Typical tasks include classification (e.g., images, music, documents), transcription, as well as the creation of original content (reviews, stories, blog posts) [Ipeirotis 2010a]. Psychologists, soci-



Design HIT Templates

Welcome! You have not created any templates yet. To get started, use one of the sample templates below. Visit the Resource Center or read the User Guide for help.

Sample HIT Templates

HIT Template Name	HIT Title	
Basic Open-ended Question	Answer a Simple Question See an example	Start with this template
Blank Template	Default Title See an example	Start with this template
Data Collection	Find the Website Address for Restaurants See an example	Start with this template
Data Correction	Provide the correct spelling of search terms See an example	Start with this template
Data Extraction	Get Product Name from Image See an example	Start with this template
Image Filtering	Flag offensive content images (WARNING: This HIT may contain offensive content. Worker discretion is advised.) See an example	Start with this template
Image Tagging	Tag an image See an example	Start with this template

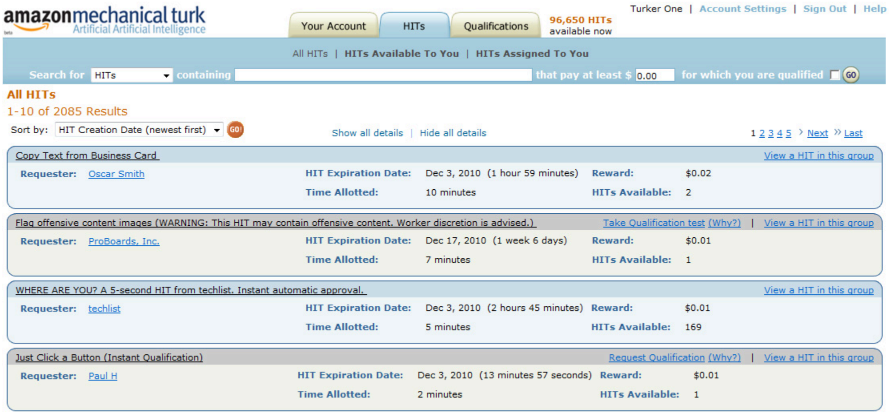
Figure 2.2: Requester interface.

ologists and economics are beginning to distribute their experiments, previously done in a laboratory, as tasks on Mechanical Turk [Mason 2010, Horton 2010, Paolacci 2010], because of the lower cost and comparable results [Paolacci 2010] as well as the access to a larger, more global and heterogeneous pool of subjects [Mason 2010]. Tasks can be created programmatically through an API, or manually based on a set of templates provided by the Mechanical Turk Requester Interface [MTurkRequestorInterface 2012], as shown in Figure 2.2.

Anyone can sign up to become a worker on Mechanical Turk. For workers, Mechanical Turk provides functionalities (see Figure 2.3) for searching for tasks by keywords and minimum payment, and allows workers to order results by the recency, number of assignments, reward amount, expiration date, title and duration of the HITs. Upon the completion of a task and the approval of the requester, workers earn the predefined reward amount, sometimes supplemented with a bonus.

There has been research documenting the demographics of workers on Mechanical Turk using surveys [Ross 2010, Ipeirotis 2010b, Mason 2010]. In a survey conducted in 2010 of 1,000 Mechanical Turk workers [Ipeirotis 2010b], workers are found to represent 66 countries, with the majority ($\sim$ 80%) from the United States and India. The survey asked workers questions about their age, income and education level, marital status, household size, and their experiences on Mechanical Turk such as time spent per week, number of HITs completed, amount of money earned, and their primary motivation for working as Turkers.

One interesting conclusion from the survey [Ipeirotis 2010b] is that the characteristics of the workers and their motivation depend heavily on workers' cultural background. It was found that there are significantly more ($>$ 2:1 ratio) female than


amazonmechanical turk
Artificial Artificial Intelligence
Your Account
HITs
Qualifications
96,650 HITs available now
Turker One | Account Settings | Sign Out | Help
All HITs | HITs Available To You | HITs Assigned To You
GO
GO!
Show all details | Hide all details
1 2 3 4 5 › Next » Last

Search for HITs containing that pay at least $ 0.00 for which you are qualified □

All HITs

1-10 of 2085 Results

Sort by: HIT Creation Date (newest first)

Copy Text from Business Card — View a HIT in this group

Requester: Oscar Smith HIT Expiration Date: Dec 3, 2010 (1 hour 59 minutes) Reward: $0.02
Time Allotted: 10 minutes HITs Available: 2

Flag offensive content images (WARNING: This HIT may contain offensive content. Worker discretion is advised.) — Take Qualification test (Why?) | View a HIT in this group

Requester: ProBoards, Inc. HIT Expiration Date: Dec 17, 2010 (1 week 6 days) Reward: $0.01
Time Allotted: 7 minutes HITs Available: 1

WHERE ARE YOU? A 5-second HIT from techlist. Instant automatic approval. — View a HIT in this group

Requester: techlist HIT Expiration Date: Dec 3, 2010 (2 hours 45 minutes) Reward: $0.01
Time Allotted: 5 minutes HITs Available: 169

Just Click a Button (Instant Qualification) — Request Qualification (Why?) | View a HIT in this group

Requester: Paul H HIT Expiration Date: Dec 3, 2010 (13 minutes 57 seconds) Reward: $0.01
Time Allotted: 2 minutes HITs Available: 1

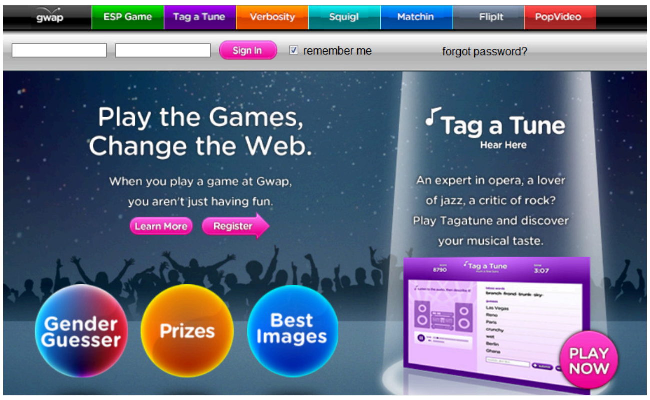
Figure 2.3: Worker interface.

male workers in the United States, while the reverse is true in India. Turkers are on average younger and have lower income than the general population. However, a much higher skew is observed amongst workers in India than the United States. Likewise, the motivations for doing tasks on Mechanical Turk are different for American and Indian workers. Workers from the United States reported their work on Mechanical Turk as a secondary source of income and as a source of entertainment; while workers from India see work on Mechanical Turk as a primary source of income. Similar findings were reported in [Ross 2010] which, additionally, analyzes the shifts in the demographics of Turkers, using six surveys conducted at semi-regular intervals within a 20 month period. They reported that Turkers seem to be getting younger, have lower income but higher education. Turkers report to earn, on average, only approximately $2.00 per hour, and some work for as many as 15 hours a week, making the work on Mechanical Turk a part-time job for some.

Gamers

It is estimated that over 200 million users, i.e., over 25% of all Internet users, play online games every week [IGDAWhitePaper 2009]. Many of the online gamers play a genre of games called *casual games*. Causal games, which have been coined "video games for the mass consumers," have several properties [IGDAWhitePaper 2009] that make them appealing to even those who do not normally consider themselves as gamers. First, casual games have low barrier to entry, e.g., they can be easily accessed online with minimum to no setup. Second, They typically have only a few simple controls, and therefore extremely easy to learn. Third, they are *non-punishing*, e.g., allowing players ample of opportunities to score. Fourth, they can be consumed within short periods of time, e.g., 5–20 minutes during work breaks. Finally, casual games are typically inclusive, gender-neutral, and contain little violent content. This makes causal games suitable for players of different ages and from all walks of life.

The idea of using causal games as a medium for computation was introduced by von Ahn and Dabbish in 2002, with the introduction of the ESP Game [von Ahn 2004]. The idea is to engage pairs of players in a simple game, where they tag images independently and are rewarded when their tags agree. Despite its simplicity, the ESP game was played by hundreds of thousands of people, rapidly generating keywords that can be used to index images to power Web image search. The ESP Game was licensed by Google in 2006, who made its own version called the Google Image Labeler [GoogleImageLabeler 2012]. Since then, there have been many human computation games developed to tackle a variety of AI problems. For example, GWAP.com (see Figure 2.4) now hosts, in addition to the ESP Game, six human computation games—TagATune (for collecting music tags), Verbosity (for collecting common sense facts), Squigl (for locating objects in images via tracing), Matchin (for ranking images by preferences), FlipIt (for measuring image similarity) and PopVideo (for collecting tags of videos).



Play the Games, Change the Web.

When you play a game at Gwap, you aren't just having fun.

Tag a Tune

Hear Here

An expert in opera, a lover of jazz, a critic of rock? Play Tagatune and discover your musical taste.

Figure 2.4: GWAP.com.

According to the 2008–2009 Causal Games White Paper released by The International Game Developers Association (IGDA) [IGDAWhitePaper 2009], gamers play causal games in order to "relax, pass time, socialize, or achieve certain goals and challenges," and typically do not see themselves as being "gamers." In terms of demographics, it was found that 74% of causal gamers are female between the ages of 30 to 45 years old, playing mostly puzzle, word and card games. Currently, there exists no equivalent survey on the demographics of gamers who are attracted *specifically* to human computation games; this information would be invaluable.

2.1.2 Games with a Purpose

There are major differences between how humans and machines compute. While one can safely assume that machines will give a consistent (i.e., always the same) answer, unlike machines, the same question asked to *human computers* might yield different answers that are biased by their particular competence and expertise, intentions, interpretation of the question, personal preferences and opinions, and general physical and psychological limitations, e.g., fatigue, lack of motivation and cognitive overload. Quality control, therefore, is a central challenge in human computation.

Within a human computation system, there are several points of intervention for quality control. For example, *before* computation takes place, the system can route tasks to the most competent worker. Likewise, *after* computation, the system can determine if the output is correct by aggregating redundant, but independent, answers, or by filtering out the bad outputs. Ultimately, high-quality outputs are not achievable if the human computers are unmotivated or unwilling to tell the truth in the first place. One way to elicit truthful responses from workers is to design a set of rules for interacting with the system in which workers benefit the most by being truthful. This set of rules, referred to as a *mechanism* [Jackson 2003, Nisan 2007], defines the set of permissible actions for the worker and specifies how the final outcome will be computed based on those actions. These mechanisms are safeguards that are placed *at* the time of computation, instead of before or afterwards.

Our use of the term *mechanism* is borrowed from a field of research called *mechanism design* [Nisan 2007], which studies systems in which multiple self-interested agents hold private information (that we want revealed truthfully) that is essential to the computation of a globally optimal solution. Because each participant is considered *rational*, i.e., with the selfish goal of maximizing one's own expected payoff, he or she may want to withhold or falsify information. In order to achieve the best economic outcomes, the goal of the system designer is to find a set of rules in which participants benefit the most by sharing their private information truthfully. A canonical example is the Vickrey-Clarke-Groves (VCG) mechanism—when used in a single-item auction, the mechanism specifies that the agent with the highest bid should receive the item, but charged the price of the second highest bid. It has been shown that the VCG mechanism is *incentive-compatible* [Hurwicz 1972]—the bidders cannot do better by mis-reporting their true valuation of the item. An important premise behind mechanism design is that agents cannot be instructed, taught or forced to behave in a certain way; however, by designing rules that align the motivation of the agents with the objectives of the system, we can encourage agents to report their private information truthfully.

In human computation, the term *mechanism* takes on a similar but slightly different meaning. Here, the private information that the human computers hold, and that our system wants to collect, is the *true* output to a computation task. We design mechanisms—a set of rules governing how the output of each human computer will jointly determine an outcome (i.e., reward or penalty)—in order to incentivize human computers to produce outputs in a truthful way. In this section, we will show how these mechanisms can be embedded in multi-player online games. Games are a particularly powerful vehicle for computation as they have the potential to reach a huge number of willing participants over the Web.

The first human computation game, the ESP Game [von Ahn 2004] (Figure 2.5), was created to collect tags that can be used to describe and index images, making them easily retrievable on the Internet. In this game, two players are given the same image, and asked to independently enter tags that describe that image. Upon agreeing on a tag, the players are rewarded with points and the image is successfully labeled.

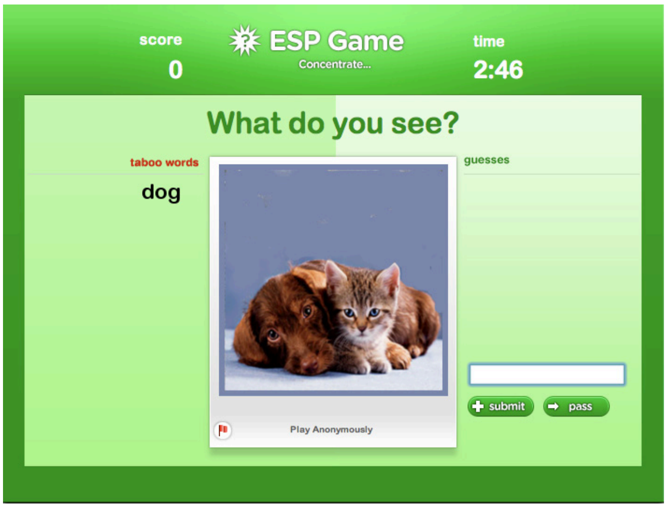
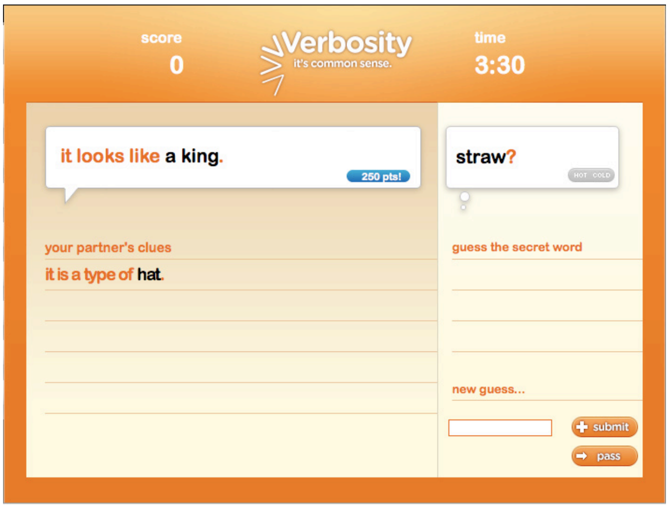


Figure 2.5: The ESP game.

To motivate truthful responses, the ESP Game combines three ideas—independence, agreement and shared information. By having two players independently generate the same tag, the system has higher confidence that the tag is correct than if the tag is generated by a single person. Furthermore, the only common information that the two players share is the image; in the absence of extra information (i.e., assuming that players do not communicate with each other), players are more likely to find a matching tag if they limit themselves to only the tags that are relevant to the image, a search space that is much smaller than the set of all words in the English language. The ESP Game is a specific instance of a mechanism called "output-agreement," where two players get the same input and are rewarded when their outputs agree.

Mechanisms are generalizable—in fact, the output-agreement mechanism has been successfully applied to other problems, including image preference [Bennett 2009, Hacker 2009], music classification [Mandel 2009b, Turnbull 2007b], ontology construction [Siorpaes 2008, Vickrey 2008] and sentiment analysis [Seemakurty 2010]. In Matchin [Hacker 2009], for example, two players are shown a pair of images and asked to vote for the one they think their partner will prefer. They are rewarded with points if their votes match. A global ranking of image preferences can then be derived from the aggregate votes. Another example is Squigl [Lee 2009], a game for gathering segmentation data for images in which two players are shown the same image and an associated label, then are asked to draw an outline around the object in the image with that label. Points are awarded based on how much the two outlines of the object overlap. In PictureThis [Bennett 2009], players are shown a label and a list of images and asked to select the image that is the most relevant to that label. Players are again rewarded if their selections match. The output-agreement mechanism has also been extended to games for knowledge extraction, such as Ontogame and Ontotube [Siorpaes 2008],



it looks like a king.

straw?

it is a type of hat.

Figure 2.6: Verbosity.

in which players are given various types of input objects (e.g., Wikipedia excerpts, YouTube videos, eBay auctions) and an ontology, then asked to annotate the input object using the given ontology. In all of these games, the reward system is the same as the one originally introduced in the ESP Game: matching on the output.

Leveraging Communication

There are some games that allow for communication between partner. An important class of games that allow for open communication is called *function computation* mechanisms, where players are given some partial input for which they need to perform some computation in order to compute an auxiliary function which determines the reward of both players.

Verbosity [Speer 2010] and Peekaboom [von Ahn 2006a] are two examples of human computation games that use an asymmetric version of the *function computation* mechanism to elicit truthful outputs from players. Verbosity [von Ahn 2006b] (Figure 2.6) is a game for collecting common sense facts. In this game, players alternate between the role of a *describer* and *guesser*. The describer is given a secret word (e.g., "crown") which he has describe to his partner, the guesser, by revealing clues about the secret word (e.g., "it is a kind of hat"). Both players are rewarded if the guesser is able to guess the secret word. The mechanism requires the guesser to compute the auxiliary function "what is the secret word" given the outputs (i.e., clues) of the describer. It is asymmetric in the sense that only one of the players is responsible for computing this auxiliary functional and that the communication is (mostly) unidirectional.

Another example of an asymmetric function computation game is Peekaboom (Figure 2.7). Peekabom is a game for locating objects in images and involves two players—the *boomer* and the *peeker*. The boomer is given an image and a secret

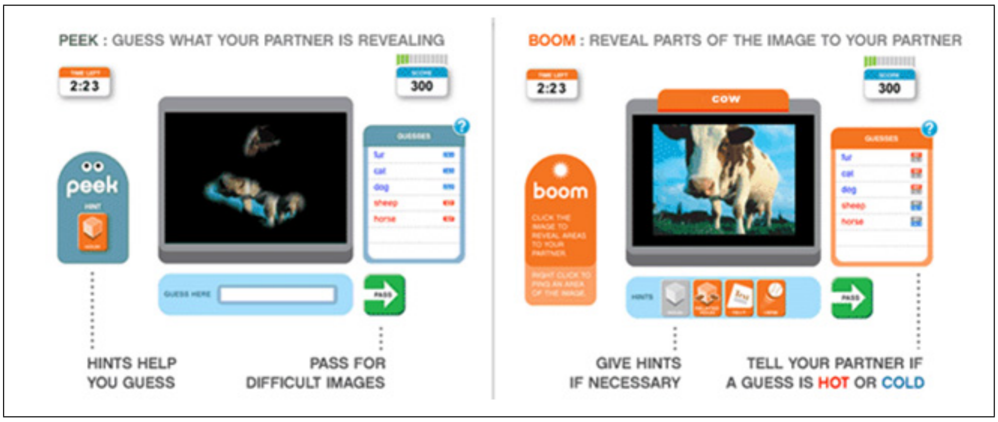

PEEK : GUESS WHAT YOUR PARTNER IS REVEALING
2:23
300
peek
fur
cat
dog
sheep
horse
HINTS HELP YOU GUESS
PASS FOR DIFFICULT IMAGES
BOOM : REVEAL PARTS OF THE IMAGE TO YOUR PARTNER
2:23
300
cow
boom
CLICK THE IMAGE TO REVEAL AREAS TO YOUR PARTNER
RIGHT CLICK TO PING AN AREA OF THE IMAGE
fur
cat
dog
sheep
horse
HINTS
PASS
GIVE HINTS IF NECESSARY
TELL YOUR PARTNER IF A GUESS IS HOT OR COLD

Figure 2.7: Peekaboom.

word (e.g., the word "cow) and must click on and reveal the part of the image associated with the secret word to his partner, the peeker. On the other hand, the peeker is initially given a blank image that is slowly unveiled by the boomer, and must guess the secret word as quickly as possible. Both players are rewarded when the peeker guesses the correct secret word.

In both Verbosity and Peekaboom, the auxiliary function asks "what is the secret input object your partner holds, given his or her descriptions of that object?" Asymmetric function computation mechanisms with this type of auxiliary functions are referred to as *inversion problem* mechanisms [von Ahn 2008b].

Designers of human computation games are faced with the challenge of building a system that simultaneously meet two (and often competing) objectives—to satisfy the players and to perform efficient and accurate computation. On the one hand, one can design a fun game that attracts a lot of players, but which does not collect any useful data. On the other hand, if the game is designed to collect the cleanest possible data, without paying attention to the enjoyability of the task, then no player would be interested.

In function computation games, e.g.,Verbosity and Peekaboom, this tradeoff is apparent. Granting players more freedom of expression (e.g., allowing them to enter free form text, or communicate with each other) can make the game more entertaining, but can lead to noisy data that requires a great deal of post-processing. In Verbosity, a significant amount of filtering needs to be done to the collected data before they are considered trusted [Speer 2010]. This is because the describers sometimes cheat by entering clues that are not common sense facts, but shortcuts for revealing the answer to the guesser [Speer 2010]. Common cheats include clues about the number of letters in the secret word (e.g., "it has three letters"), or mnemonics (e.g., "it sounds like king").

A Brief Survey of Games and Mechanisms

There has been a large number of human computation games invented to tackle different problems, many of them are identical in terms of the underlying mechanisms. Table 2.2 provides a summary of some existing human computation games and their underlying mechanisms; consult [Thaler 2011] for a comprehensive survey of human computation games for knowledge acquisition.

Table 2.2: Survey of human computation games.

Game	**Description**	**Mechanism**	**AI Problem**
The ESP Game	two players match on a tag for the same image	output-agreement	object recognition
Peekaboom	player 1 reveals parts of the image associated with a secret word, player 2 must guess the secret word	function computation (problem inversion)	object location
Verbosity	player 1 describes the properties of the entity associated with a secret word, player 2 must guess the secret word	function computation (problem inversion)	knowledge extraction
FoldIt	players fold protein structures to minimize total energy	function computation (optimization)	protein folding
HerdIt	players select tags that describe the music	output-agreement	music classification
Categorilla	players name an entity that fits a template (e.g., Things that fly)	output-agreement	natural language processing
MoodSwings	players click on a 2-dimensional grid to indicate the valence and intensity of the mood of a music clip	output-agreement	music mood classification
Phrase Detective	players identify relationships between words and phrases in a short piece of text	output-agreement	natural language processing
Phylo	players align colored blocks by moving them horizontally and inserting gaps	output-agreement	genome alignment

2.2 Machine Computation

2.2.1 Music

One of the key challenges in music information retrieval is the need to quickly and accurately index the ever growing collection of music on the Web. There has been an influx of recent research on machine learning methods for automatically classifying music by semantic tags, including Support Vector Machines [Li 2003, Mandel 2005], Gaussian Mixture Models [Turnbull 2007a, Turnbull 2008b], Boosting [Bertin-Mahieux 2008a], Logistic Regression [Bergstra 2006a], and other probabilistic models [Hoffman 2009]. The majority of these methods are supervised learning methods, requiring a large amount of labeled music as training data, which has been traditionally difficult and costly to obtain.

There is now a proliferation of online music websites, where millions of users visit daily, providing an unprecedented amount of useful information about each piece of music. For example, collaborative tagging websites, such as Last.FM, collects on the order of 2 million tags per month [Lamere 2008a]. Without prompting, human users are performing meaningful computation each day, mapping music to tags. There are a variety of ways to obtain tags for music, e.g., conducting a survey, harvesting social tags, through the use of human computation games, and mining web documents [Turnbull 2008a]. Recently, the Million Songs Dataset was created from mining the Web. It should be noted that the mined music tags are largely associated at the artist-level or album-level, and not at the song-level.

2.2.2 Images

Object Recognition is a topic in Computer Vision where there has been substantial work [Grauman 2011]. Beyond object recognition, there has been a recent movement towards using an intermediate layer of human-understandable attributes for classification. Instead of learning a classifier to map images features to classes (e.g., 'dog"), one can map image features first to a set of semantic attributes (e.g., 'has four legs", 'is furry", 'is brown") and then map the predicted semantic attributes to the class with the most similar set of attributes. This method is scalable – since many objects in the world can be succinctly described using only a small number of semantic attributes, learning to map low level features to this efficient code can allow instances to be classified into new categories where no training examples is available. This idea of zero shot learning has been studied in the context of thought prediction using fMRI images [Palatucci 2009] and visual object recognition [Farhadi 2010, Farhadi 2009, Kumar 2009, Lampert 2009] The second benefit is explanatory power: the learning system can now describe to users the reasons behind its predictions.

With the exception of [Parikh 2011], most previous works study the feasibility of image classification using an intermediate layer of a fixed set of manually curated attributes. For example, Kumar et al. [Kumar 2009] manually created 65 attributes for face recognition, and paid $5000 to obtain attribute values from workers on Mechanical Turk. The Animal with Attributes dataset was created using the 50 attributes proposed in [Kemp 2006]. The outdoor scene datasets provided by [Farhadi 2009, Farhadi 2010] uses a fixed set of 64 attributes, describing the objects' shape (e.g., 'cylindrical"), parts (e.g., 'has window") and material (e.g., 'is shiny"). On the other hand, there has also been recent work on using text corpus to automatically characterize the visual attributes of objects [Berg 2010, Rohrbach 2010] without any human supervision.

2.2.3 Named Entities

In text mining, there are now many systems that can automatically extract facts about real world entities from the Web. There has been a large amount of work on relation learning. For example, NELL (which stands for Never-Ending Language Learner) [Betteridge 2009, Carlson 2009, Carlson 2010b] is a system that can continuously extract instances of categories and relations from the Web to populate a structured knowledge base (i.e., an ontology). Using a semi-supervised approach [Blum 1998, Carlson 2010b, O. Chapelle 2006] on hundreds of millions of webpages, the system iteratively learns assertions (e.g., the names of individual athletes, what sport they play, their team, which stadium and city the team plays in, who their coach is) by discovering text patterns associated with particular categories (e.g., the text string 'sports figures like X" suggests that X is an athlete) and relations (e.g., 'X superstars such as Y" suggests that athlete Y plays sport X). The current version of the system ([Carlson 2010b, ReadTheWeb 2012]) has already learned to extract a knowledge base containing over a million assertions with an accuracy around 85% and over 10 million additional assertions in which it has lower confidence. The eventual goal of NELL is to be able to continuously learn to read, i.e., from the same text corpora, 'extract more information more accurately" [Carlson 2010a] than the day before. In the context of NELL, we use the word attributes to refer to either categories (e.g., 'dog", 'athlete") or relations (e.g., 'has tail", 'plays football"). Other similar *open information extraction systems* include TextRunner [Banko 2007] and WOE [Wu 2010].

2.3 Joint Human-Machine Computation

Most previous work on systems that combine humans and machines focus on the role of machines as an optimizer – to help improve the accuracy and efficiency of human computation algorithms, or help reduce the cost of human computation by choosing only informative questions to ask.

Human-in-the-loop systems are automated systems that involve humans to perform part of their functions, in order to overcome sensing and reasoning limitations. This is prevalent in robotics, where the performance of automated systems is often not deemed adequate to handle real world situations in the failsafe manner [Hearst 1999, Rosenthal 2010]. An important subclass is human-in-the-loop learning systems, which are systems that are capable of self improving given human feedback during the learning or execution process. Humans can act as coaches to the system and teach the system what to learn, e.g., by providing training labels [Settles 2012, Lewis 1994], feature values [Maytal 2009], reward signals [Thomas 2006], target controlled policies [Abbeel 2004] and information about hidden states [Kapoor 2008]. Alternatively, humans can also act as critics, by providing evaluative (e.g., is the answer correct or not) and corrective (e.g., the right answer is X) feedback to the learning system [Culotta 2006, Shilman 2006].

One important dimension of human-in-the-loop learning systems is whether tasks are assigned using a push (where the machine needs to actively seek for human help) versus pull (where the humans seek for tasks to perform) model. If tasks are assigned using the push model, then it matters greatly that the system can accurately predict whether humans are available, willing to help, and do not mind being interrupted from their current activities. This is important whether the machine is trying to help the humans with their tasks (e.g., schedule a meeting, guide notifications, etc.) [Kapoor 2007] or perform its own tasks (e.g., finding a room) [Rosenthal 2010]. In contrast, under the pull model, a system will not give users tasks unless they ask for one. Here, the feedback is often implicit, with humans performing an activity while the machine is observing in the background. For example, search engines can retrieve relevant documents for human users, whose clicks are noisy indication of whether the retrieved results are relevant or not [Joachims 2005]. Human computation systems, for example, use the pull-based model – by aligning system and user interests, people will come voluntarily and ask for tasks, which the system needs to hand out whether it is ready or not. On the other hand, human computation systems need not concern themselves with issues such as availability and interruptibility.

An effective human computation system should be able to interweave machine and human capabilities seamlessly. This idea is not new; many research concepts familiar to the AI community, such as complementary computing [Horvitz 2007b], mixed-initiative systems [Horvitz 2007a] and interactive machine learning [Yue 2009, Fogarty 2008, Shilman 2006, Kapoor 2012], address similar questions. There are already several human computation projects where we see hybrid solutions emerging. CrowdPlan [Law 2011b] asks workers to decompose high-level search queries (in the form of missions such as “I want to quit smoking”) into a set of goals and transform the goals into search queries, leaving the actual search task to machines. Shahaf et al. [Shahaf 2010] introduces the use of machine intelligence to manage human computers as a resource, by taking into account competencies, availabilities, and payment. Branson et al. [Branson 2010] introduces a computer vision algorithm for classifying bird images, that upon facing uncertainties, asks Turkers to answer questions (e.g., “is the belly red”) to refine the answer. To compute a crowd kernel (i.e., a similarity matrix over a set of objects), Tamuz et al. [Tamuz 2011] uses an algorithm to adaptively select maximally informative triplets of objects to query for human similarity judgments, thereby approximately the true answer with fewer number of queries. Another example of joint machine (machine vision) and human computation (human classification) is the work on CrowdSynth – where human input and machine-recognized features of galaxies in sky survey are combined via models learned with machine learning [Kamar 2012].

2.3.1 Active and Proactive Learning

Active learning is a machine learning model in which the learner intelligently chooses the data from which it learns [Settles 2012]. This typically involves a iterative process, with the learner alternating between querying (asking an oracle a question about the data) and updating (incorporating the answers into the current model).

Most active learners use a supervised learner as its underlying learning algorithm and queries for the label of an instance or a batch of instances. There exist many different selection strategies for choosing instances to query for information – e.g., selecting instances whose labels have the most uncertain or ambiguous classification (uncertainty sampling) [Lewis 1994], are disagreed upon by the most experts (query-by-committee) [Seung 1992], or have the most effects on improving the generalization power of the classifier (Expected Error Reduction) [Roy 2001] etc. The performance of the active learner is judged by how quickly it can improve the classification accuracy, compared to the baseline performance of random selection. In this section, we will discuss two ways in which active learning in the human computation setting diverges from the traditional model and mention some related works.

Active learning typically assumes a hypothetical existence of a single, perfect, omniscient oracle. This assumption breaks down as we move towards a framework with human computers as oracles, who may be imperfect, unreliable and reluctant to answer. These issues have been explored extensively in Proactive learning [Donmez 2009a], which examines instance selection strategies when there are many imperfect oracles. For example, Pinar et al. [Donmez 2008] studied the problem of how to make as few queries as possible, while obtaining data from the best oracle to train a classifier using a fixed budget. The algorithm has a discovery phases for probing the characteristics of each worker and a task assignment phase in which the (task, worker) pairs are chosen to maximize the cost-benefit tradeoffs. They model several types of workers, including reliable (who always answer) versus unreliable workers (who sometimes refuse to answer), faillable versus infallible workers, and workers with uniform costs versus those with costs that vary across tasks.

Instead of the two phase procedure (i.e., used in [Donmez 2008]) of first estimating utilities of each worker, then performing the task assignments to maximize the estimated utilities, there are algorithms that interweave the estimation and task routing process. At every time step, the algorithm can decide whether to explore, i.e., assign an information gathering [North 1990] task to learn about a worker's characteristics, versus exploit, i.e., assign a task to the worker that the system currently believes is best. A particular online algorithm for assigning tasks to multiple oracles with variable reliability is discussed in [Donmez 2009b]. The idea is to adapt the Interval Estimation (IE) algorithm for selecting oracles (each with different level of competence) for a labeling task. Interval Estimation (IE) Learning [Kaelbling 1993, Kaelbling] is a technique for choosing actions (e.g., the action of assigning a task to a particular worker) that balances the exploration and exploitation tradeoff. For each action a_i, the IE algorithm keeps tract the number of times n_i the action has been executed and the number of times w_i that the execution was successful. At each time step, the algorithm estimates the confidence interval of the success probability of each action, and chooses the one with the highest upper bound. The upper interval value can be large either due to a high sample mean of the success probability, or due to the uncertainties in our estimates. As more actions were performed, the interval shrinks and the algorithm is able to then select the best workers for any task. In the particular adaptation of the IE algorithm [Donmez 2009b], called IEThres, the goal is to minimize the number of queries and filter out the unreliable oracles early in the process. IEThres was also used in [Donmez 2010a] to select oracles with time-varying accuracy – the idea being that the accuracy of human workers is likely to change over time, becoming less accurate as they are fatigued or get bored, or more accurate as they gain skills and knowledge by performing particular tasks. In [Donmez 2010b], a sequential Bayesian model was used to estimate the accuracy of a worker at time t based on previous observations; based on these accuracy estimates, IEThres was then used to select the best human oracles for the task. This idea of using variance to indicate which worker needs to be explored is also used in the online EM approach proposed by [Welinder 2010].

2.3.2 Closed versus Open World

Many AI and machine learning algorithms make the closed world assumption (i.e., what is not known to be true must be false) and the closed domain assumptions (i.e., there are no other objects in the world except for the ones the system knows about). Learning algorithms typically assume that objects can be classified into a fixed set of classes and represented by a fixed set of features and feature values. For example, several music tagging algorithms trained on the data extracted by human computation games [Turnbull 2007b, Mandel 2009b] assume that the absence of a tag for a given music clip means that it is irrelevant for that clip (i.e., the absence of the 'happy" tag means that the music is sad).

In the real world, these assumptions rarely hold. For example, when players are presented with images to process in a human computation game, they may come up with a tag that already exists in the current vocabulary, or a new one that the system has never seen before. Consider Horvitz's characterization of the open world problem [Horvitz 2008]: 'The open-world assumption is the assumption that the truth value of a statement is independent of whether or not it is known by any single observer or agent to be true. I use open world [more broadly] to refer to models of machinery that incorporate implicit or explicit machinery for representing and grappling with assumed incompleteness in representations, not just in truth-values."

An active learner that acquires feedback through a human computation system must deal with the open world problem, i.e., it must decide when it has incomplete knowledge of the world, and ask humans to help extend its representation. For example, an active learner for NELL needs to decide when to acquire new categories and relations to add to its ontology, and when to simply ask for humans to help evaluate and correct its current beliefs. The active learner must also have some notion of confidence about the beliefs in the system, using which to detect incorrect beliefs even when the system strongly believes them to be true.

Chapter 3

Learning Attributes in Vocabulary-Rich Settings

3.1 Overview

Querying by semantic tags is arguably one of the most intuitive methods for retrieving music. However, until recently [Bertin-Mahieux 2008b, Chen 2009, Hoffman 2009, Turnbull 2008b], most retrieval methods focused on querying by metadata [Whitman 2002] (e.g., artist or album names), similarity [Goto 2004], humming [Dannenberg 2004], beatboxing [Kapur 2004] and tapping [Eisenberg 2004], or using a small, fixed set of categories (e.g., genre [Tzanetakis 2002, Tzanetakis 2001], mood [Trohidis 2008], or instrumentation [Herrera 2003]) as keywords. Retrieval by semantic tags is still not a prevalent music retrieval strategy, because there is a large amount of music on the Web that remained untagged, and music tagging algorithms are not yet accurate enough to be useful. Yet, Music tags are incredibly valuable – they enable users to browse, organize, and retrieve music in an semantic way.

One potential source of music tags is collaborative tagging websites, such as Last.FM, which collect on the order of 2 million tags per month [Lamere 2008a] from tens of thousands of users. However, there are two known issues with using such "social tags" as labeled data for multimedia objects. First, only the popular items are typically tagged, leaving a large proportion of the multimedia objects on the Web untagged [Lamere 2008a]. Second, for multimedia objects with a time component, such as sound, music, and video clips, social tags found online often describe the object as a whole, making it difficult to link tags with specific content elements. This makes social tags unsuitable as data for training algorithms for music and video tagging, which rely on specific content elements being tagged (as opposed to the overall content). Other approaches, such as human computation games, became an attractive alternative method for collecting music tags.

While objects in images can typically be referenced in a limited number of ways, music can be described by a vast and diverse vocabulary. In this chapter, we investigate how to design an effective human computation system for learning attributes in such vocabulary-rich settings.

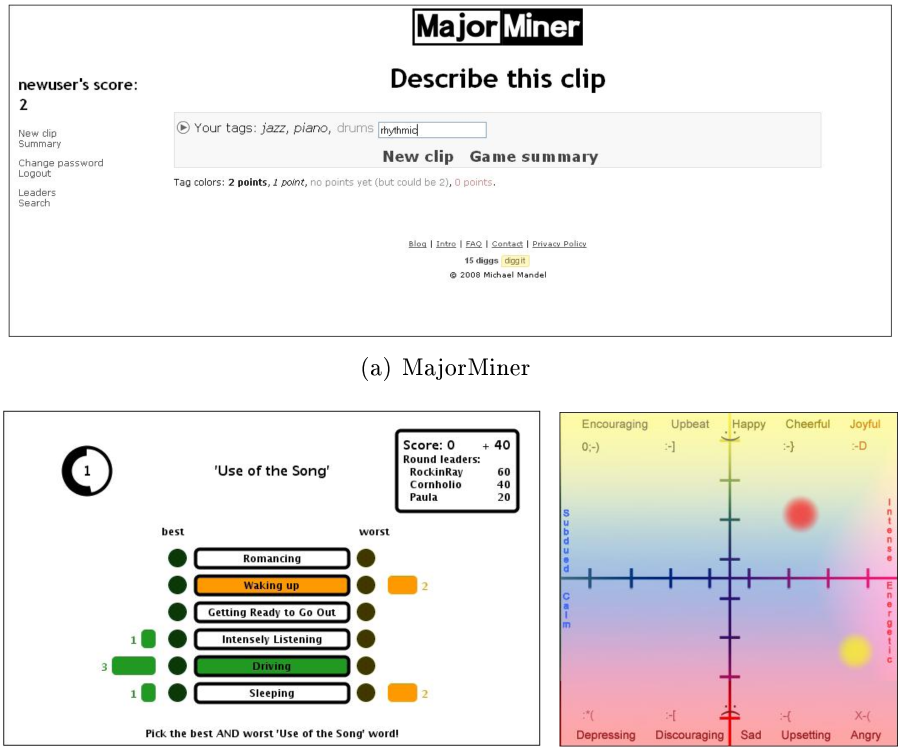
3.2 Game Mechanism for Learning Music Attributes

3.2.1 Motivation

Games with a purpose, as we mentioned in the previous chapter, have proven to be an effective method for extracting attributes from game players. The ESP Game mechanism, in particular, was hugely successful: millions of image tags have been collected via the game, and a few years after its deployment, the game is still visited by a healthy number of players each day. This game mechanism has been re-used as the design template for many other games, including games for music annotation. By 2007, there are three human computation games for music annotation that follow the ESP Game (or output-agreement) mechanism (see Figure 3.1), namely Major-Miner [Mandel 2009b], the Listen Game [Turnbull 2007b], and MoodSwings [Kim 2008]. MajorMiner [Mandel 2009b] is a single-player game in which players are asked to enter descriptions for ten-second music clips. Players receive points for entering tags that agree with tags that were previously entered for the same music clip. The scoring system encourages originality by giving a player more points to be the first to associate a particular tag with a given music clip. The Listen Game [Turnbull 2007b] is a multiplayer game in which players are asked to describe 30-second music clips by selecting the best and worst tags from six choices. In the "freestyle" rounds, players can suggest new tags for a clip. Players are rewarded based on agreement and response speed. Finally, MoodSwings [Kim 2008] is a game for annotating the mood of a given piece of music in which players are asked to indicate the mood, in terms of arousal and valence, by clicking on a two-dimensional grid. Players are given points for agreeing with each other in terms of the proximity of their mouse clicks. All of these games use variants of the output-agreement mechanism.

The output-agreement mechanism, however, has serious shortcomings. It has been noted that the image tags produced by the ESP Game tend to be common and uninformative [Weber 2009, Jain 2008]. This is a direct consequence of the output-agreement mechanism—needing to agree with his partner, a player's best strategy is to enter common tags that are likely to be entered by any person. A possible remedy is to use rewards to motivate players to enter more specific tags—e.g., the Google Image Labeler gives players higher scores for more specific labels. Another solution is to impose restrictions on what the players are allowed to enter. In the ESP Game, *taboo words* [von Ahn 2004] are introduced to prevent players from re-entering high frequency tags. None of these approaches seem to solve the problem completely [Weber 2009]. In fact, in many output-agreement games, the improper use of restrictions can lead to bad results. For example, in the effort to collect a diverse set of results, the game Categorilla [Vickrey 2008] forces players to enter only an answer that begins with a particular letter, without knowing if such an answer actually exists. As a result, players often have great difficulty coming up with such word, and end up generating nonsensical answers instead.

The research covered in this chapter answers the following questions: Is output-agreement an effective mechanism for extracting music attributes? If not, what



newuser's score:
2

Your tags: *jazz*, *piano*, drums rhythmic

Tag colors: **2 points**, *1 point*, no points yet (but could be 2), 0 points.

© 2008 Michael Mandel

(a) MajorMiner

Score: 0 + 40
Round leaders:
RockinRay 60
Cornholio 40
Paula 20

Pick the best AND worst 'Use of the Song' word!

(b) The Listen Game

(c) MoodSwings

Figure 3.1: Output-agreement human computation games for collecting music data

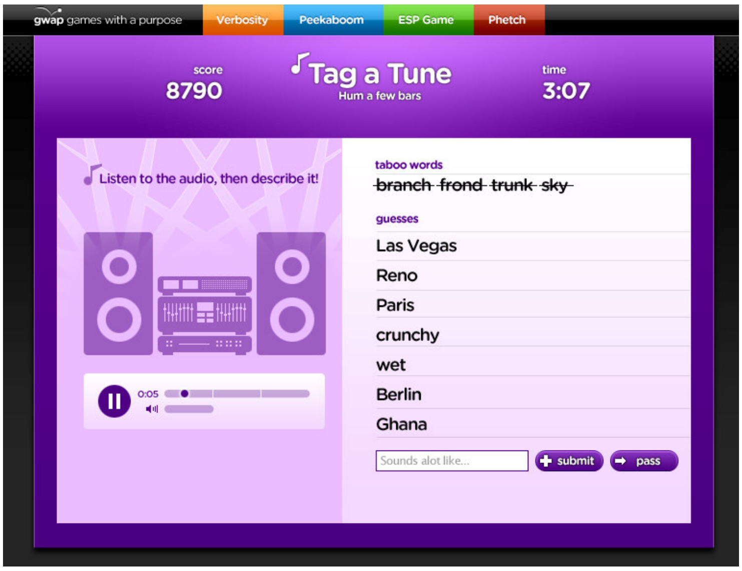
are the alternatives? How would a different game mechanism impact the system's ability to achieve its *task-centric objective*, i.e., to collect music tags that are useful for training automated tagging algorithms? In the following sections, we will discuss our findings for each of these questions in turn.

3.2.2 Evaluation of the Output-Agreement Mechanism

Is output-agreement an effective game mechanism for extracting music attributes? To begin investigating this question, we built a prototype music annotation game called TagATune, which used the output-agreement mechanism. In the prototype game [Law 2007], two players were presented with 30-second audio clips and asked to type descriptions for them (see Figure 3.2).

The initial prototype served sounds only (not songs). Players were rewarded when descriptions matched. "Taboo words" [von Ahn 2004] were also used to encourage players to enter new tags. Although the prototype game was able to collect semantically meaningful tags, the average enjoyability rating was only 3.4 out of 5, based on a survey submitted by 54 participants in a user study [Law 2007]. Moreover, it was found that 36% of the time, players opted to pass instead of entering a description [Law 2007].

There were two additional opportunities for gathering informal observations on



Listen to the audio, then describe it!

taboo words

~~branch frond trunk sky~~

guesses

Las Vegas

Reno

Paris

crunchy

wet

Berlin

Ghana

Figure 3.2: TagATune prototype

.

the TagATune prototype game: a game demo session at the ISMIR 2007 Conference and a human computation workshop for elementary school students (held at Creative TechNight, a weekly event run by Carnegie Mellon School of Computer Science to foster young girls' interest in technology). In both game-playing sessions, the key observation was that players were often frustrated by being unable to match on a tag. Specifically, players often entered tags that meant the same thing, but that were expressed differently (e.g., 'cars on a street' versus 'traffic'). Moreover, since players were not allowed to communicate with each other (this requirement of output-agreement games safeguards against cheating), players found no good strategies to produce tags that match, except to enter tags that were as general as possible (e.g., 'music,' 'classical') or to rely on random chance.

The game design strategies used in MajorMiner and the Listen Game reflect this underlying problem. Because it is difficult for two players to match on a tag, MajorMiner instead uses the agreement between a player and all previous players, while the Listen game uses the agreement among a group of players for a small predefined set of tags. There are disadvantages to such design approaches: having people play by themselves eliminates the social aspects of online games and limiting players to a predefined set of tags may make the game significantly less enjoyable and useful.

The main problem with using the output-agreement mechanism to collect data for audio clips is that it can be very difficult for two players to agree on a description. Unlike images, which often contain only salient objects that can be described using words that are commonly known to people, music can be described in a wide variety of ways – e.g., by abstract concepts, such as “temperature” (e.g., chilly, warm), mood (e.g., dark, angry, mysterious), or the image it evokes (e.g., busy streets, festival),

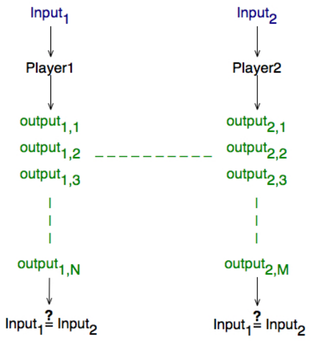


Figure 3.3: Input-agreement mechanism

.

as well as categorizations that have no clearly defined boundaries (e.g., acid-jazz, jazz-funk, smooth jazz) – and the language used to describe the different aspects of music may not be as standardized. The difficulty with arbitrary sound clips is even more marked, since the content is not always readily recognizable. In other words, music and sound clips have high *description entropy*, making it difficult, if not impossible, to match on any tags other than ones that are simple and general.

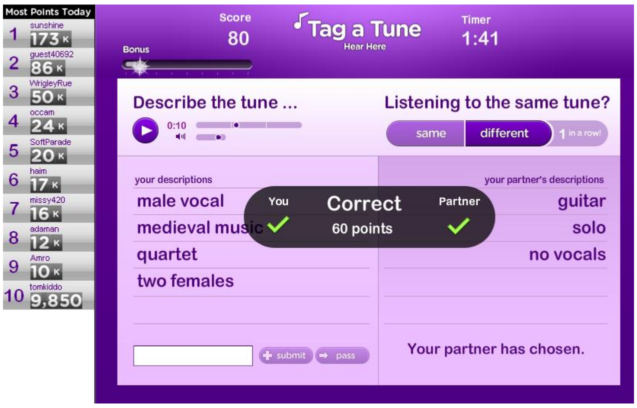
3.2.3 TagATune and Input-Agreement Mechanism

If not the output-agreement mechanism, what are the alternatives? What kind of game mechanisms can be used to collect detailed, informative tags for input objects that have high description entropy?

Our answer is a game mechanism that can motivate players to tell the truth, and yet, does not require players to match their outputs with each other. In this mechanism, two players are shown either the same music clip or different music clips and each is asked to type a description of their given music clip. Unlike output-agreement games, where all communication is forbidden, all of the players' descriptions are revealed to each other. Based on these descriptions, the players must decide whether they have been given the same music clip or not. The descriptions that players enter are exactly what we are interested in.

This mechanism belongs to a class of mechanisms called *function computation* mechanisms, where players are given some partial input (e.g., a music clip) for which they need to perform some computation (e.g., generate tags), in order to compute an auxiliary function (e.g., whether the two pieces of music are the same or different) which determines the reward of both players. TagATune employs a specific instance of the function computation mechanism called *input-agreement* [von Ahn 2008b] (Figure 3.3), where the function to compute is 1 if the input objects given to the two players are the same, 0 if they are different.

We designed a new version of the TagATune game that instantiates the input-agreement mechanism. A screenshot of the interface for a normal round of TagATune is shown in Figure 3.4.



Most Points Today

1	sunshine	173 K
2	guest40692	86 K
3	WrigleyRue	50 K
4	occam	24 K
5	SoftParade	20 K
6	haim	17 K
7	missy420	16 K
8	adaman	12 K
9	Amro	10 K
10	tomkiddo	9,850

your descriptions	your partner's descriptions
male vocal	guitar
medieval music	solo
quartet	no vocals
two females	

Correct
60 points

Your partner has chosen.

Figure 3.4: TagATune

.

In each round, two players are given either the same audio clip or different audio clips. They are provided with a basic music player interface to start, stop, and adjust the volume of the audio clip to which they are listening. Each player describes the given audio clip by typing in any number of tags, which are revealed to the partner. By reviewing each other's tags, the players decide whether they are listening to the same thing by selecting either the same or different button. After both players have voted, the game reveals the result of the round to the players and presents the next round. The game lasts three minutes in total.

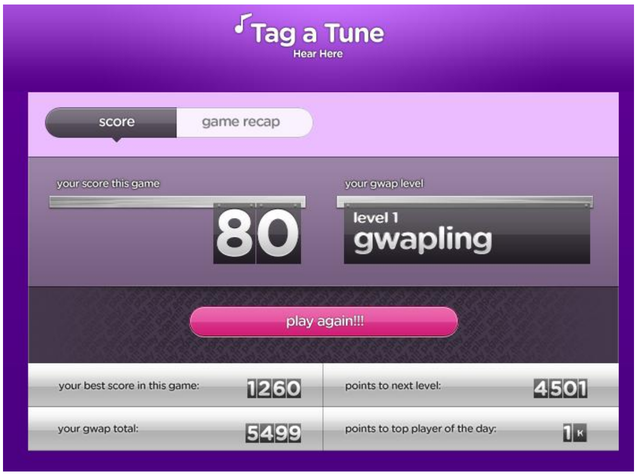
The inspiration for TagATune (and the input-agreement mechanism) comes from a psychology experiment [Healey 2007] that studies the emergence and evolution of graphical symbol systems. The experiment involved a music drawing task, where pairs of participants were given a 30-second piece of piano music and were asked to draw on a shared virtual whiteboard. Based on the drawings, the players had to decide whether they had been given the same piece of music. Remarkably, using just drawings – whether abstract (e.g., contours, lines, or graph-like representations) or figurative (e.g., recognizable objects, figures, or scenes) – players were able to guess correctly whether their inputs were the same.

3.2.3.1 Input Data

The data served to the players consists of 56,670 short (30 second) music clips from Magnatune.com and 28,715 sound clips from the FreeSound Database (http://freesound.org). Broadly speaking, the genres of music include classical, new age, electronica, rock, pop, world music, jazz, blues, heavy metal, and punk. All audio clips are provided under the Creative Commons License, allowing for much less restrictive usage than other typical music licenses. This allows audio files to be freely distributed to the public and greatly facilitates research on the data collected by the game. Moreover, the use of less well-known music minimizes the possibility that players will recognize the actual song or artist and simply describe the audio clip using tags that are already known. Finally, the shorter audio segments ensure that there is a more direct, though not guaranteed, link between content of the music and the descriptions provided. For each round, the audio clips are selected randomly. Because the input data to the game is a pair of audio clips, the number of all possible pairs of sound and music clips is large enough that random selection suffices to ensure that players will not encounter the same pair of input data too often.

3.2.3.2 Scoring Mechanism

TagATune is a cooperative game, as can be seen from its scoring mechanism: the players score points only if they both guess correctly whether they are listening to the same audio clip. Neither gains points if one of them guesses incorrectly. This provides a natural incentive for players to be truthful to each other, which in turn, generates labeled data that accurately describes the audio clip at hand. If TagATune were a competitive game, each player would be motivated to win against their partner, possibly by being malicious and misleading, and entering tags that did not describe the actual content of the audio clip. The consequence of this malicious behavior would be erroneously labeled data. Thus, a game that uses the input-agreement mechanism must be cooperative. The points in TagATune compound: the more rounds the players successfully win in a row, the more points the players get for each subsequent round. This is a general scoring mechanism to motivate players that is shared by most games on the GWAP.com game portal, where TagATune is deployed.



your score this game 80

your gwap level
level 1
gwapling

your best score in this game: 1260

points to next level: 4501

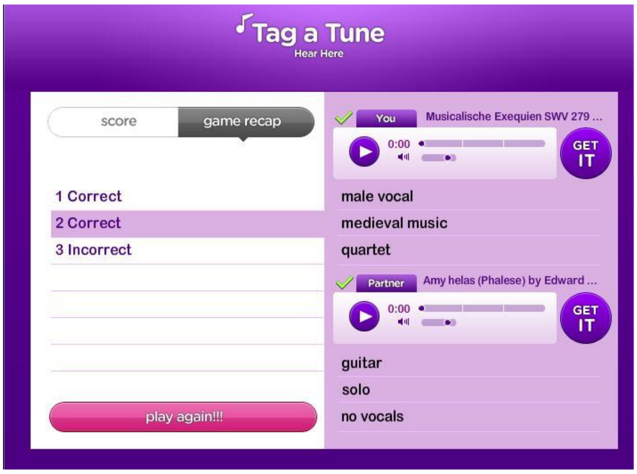
your gwap total: 5499

points to top player of the day: 1K

Figure 3.5: Score Board

.

A leader board is shown on the left of the main game panel throughout the game (see Figure 3.4). When the game is completed, a scoreboard displays the final score and the player's current GWAP level (see Figure 3.5). A GWAP level is a rank assigned to players who attain a specific number of points and each level carries a special title. The scoreboard also shows the player's best score for TagATune, the player's total accumulated score, the number of points needed to achieve the next GWAP level, and the number of points needed for the player to become the top player of the day. The leader board and scoreboard are game design elements common to all games on GWAP.com, and serve to motivate the players to strive for higher scores by playing better and more frequently.



1 Correct
2 Correct
3 Incorrect

✓ You Musicalische Exequien SWV 279 ...

male vocal
medieval music
quartet

✓ Partner Amy helas (Phalese) by Edward ...

guitar
solo
no vocals

Figure 3.6: Game Recap

.

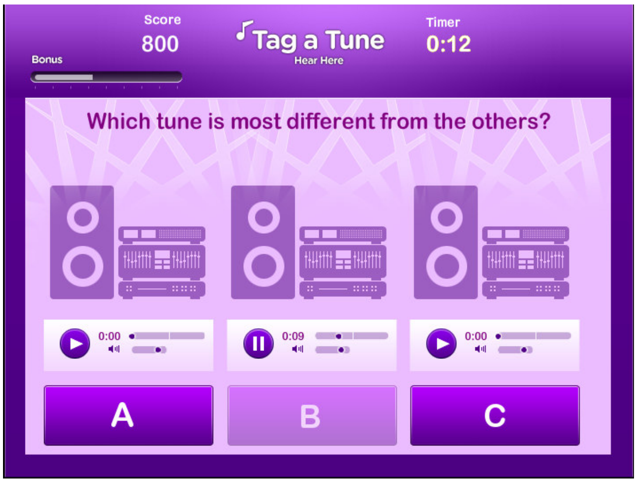
Finally, the game recap provides an opportunity for players to learn from their mistakes by reviewing their detailed performance in the game (see Figure 3.6). Players can also replay every audio clip that was presented to them during the game, and click the Get It button to download the song. This Get It functionality gives TagATune a dual purpose as a Web application for sampling new music.

3.2.3.3 Bonus Round

When the players reach 1,000 points, a bonus round is added along with an extra minute of game play. During the bonus round (Figure 3.7), players are asked to listen to three pieces of music or sound clips. Each must decide individually which one of the three clips is most different from the other two. If they agree, they both obtain points. Figure 7 shows the interface for the bonus round of TagATune.

The reason for including a bonus round is that it produces two types of additional data. First, similarity data for music is useful for powering and improving music recommendation systems. Second, the similarity between songs is potentially a good indication of the level of difficulty that a particular pair of songs would present during a normal round of TagATune. More specifically, two songs that are very similar will require a great number of more specific descriptions in order for the players to distinguish them. This similarity data can be used later to adjust the difficulty of the game and thus increase the enjoyment for the players.

This is what we can call a coupled learning [Carlson 2010b] scenario. In a normal round of the game, we are collecting data that can be used to learn a function



Which tune is most different from the others?

Figure 3.7: Bonus Round

.

$f1 : s \rightarrow \{t1, t2, ...tT\}$ that maps a given song s to its associated set of tags {t1 , t2 , . . . tT }. In addition, we are collecting data about the similarity distance between pairs of songs, i.e. a function $f2 : s \times s \rightarrow [0, 1]$, from the bonus round of the game. A closer examination reveals the relation between the two functions $f1$ and $f2$. Suppose that each song can be represented by an arbitrarily long bit string, where each bit represents a tag and is 1 if the tag describes the song and 0 otherwise. If two songs (and consequently the two bit strings) are very different, it is likely that revealing only one or two bits will enable the players to tell that the songs are different. Conversely, if two songs are very similar but different, it will take more exchange of bits for the players to arrive at the correct response. To learn $f1$, the function that maps songs to tags, the optimal solution is for the game to select, for the normal rounds, pairs of songs that are very similar yet not identical, so that the players reveal as many bits (i.e. tags) as possible before guessing whether the songs are the same or different. In other words, knowing the similarity of songs (i.e. $f2$) can help us ask the right questions to human users to learn f1. The idea of using related functions to pick examples to learn a function has not been explored in active learning, and raises interesting questions about how to select which function to learn at a given time in order to minimize the number of queries and optimize performance. In this thesis, we focus on the efficiency of TagATune in collecting high-quality attributes for music. Therefore, our analysis will be centered mainly on the results of normal rounds of TagATune.

3.2.3.4 Game Statistics

TagATune has been deployed since May 15, 2008. The statistics reported here is over the course of the first seven months since TagATune's deployment.

A total of 49,088 unique games were played by 14,224 unique players, equaling 439,760 normal rounds. Based on the statistics collected in mid-December 2008, the number of games each person played ranged from 1 to 6,286, and the total time each person spent in game play ranged from three minutes to 420 hours. The average number of games played was four. Figure 8 shows the rank-frequency curve of how many people played x number of games. The graph in Figure 3.8 almost resembles a power law: there are many people who played only a few games, and a few people who played many games. We refer to this rank-frequency curve as the player retention curve, since the curve is a useful indicator of the proportion of players who re-visited the game and the frequency of their revisits.

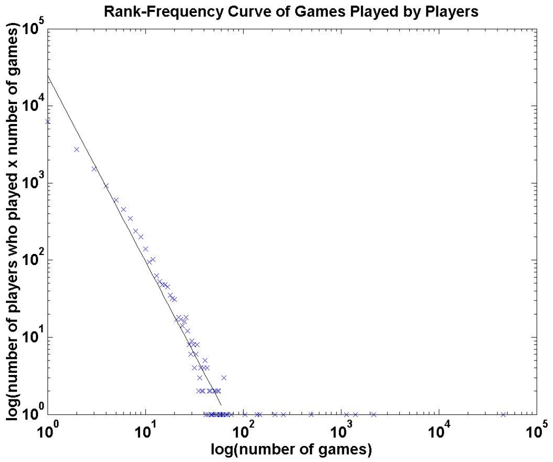


Figure 3.8: Number of people who played x number of games

.

The relative flatness of the slope of the user retention curve is a way to compare the enjoyability and popularity of different human computation games. A steep slope implies that many people played only one or a few times before abandoning the game, and not many people returned to play the game again. In contrast, a flatter slope indicates that only a few people abandoned the game after playing just a few times, while many people played a large number of games. For example, Figure 9 shows a comparison of the player retention curves for different games on GWAP.com. The results show that the player retention curves for TagATune and Squigl are similar (in terms of slopes and intercepts), indicating that the number of players who played x number of games is similar between the two games, regardless of what x is. In comparison, there are more players for the ESP Game and Matchin, for any given number of games x. Finally, when compared to other games on GWAP.com (Figure 3.9, there are substantially more players who played a large number of games of Verbosity.

Of the 439,760 rounds, players only passed on 2,203 rounds, or 0.50% of the total number of rounds. In contrast to the 36% pass rate of the prototype version of TagATune, this indicates that players are less likely to give up on a round when the new mechanism is used. In 97.36% of the rounds, both players voted same or different before the end of the round. We refer to these rounds as completed rounds.

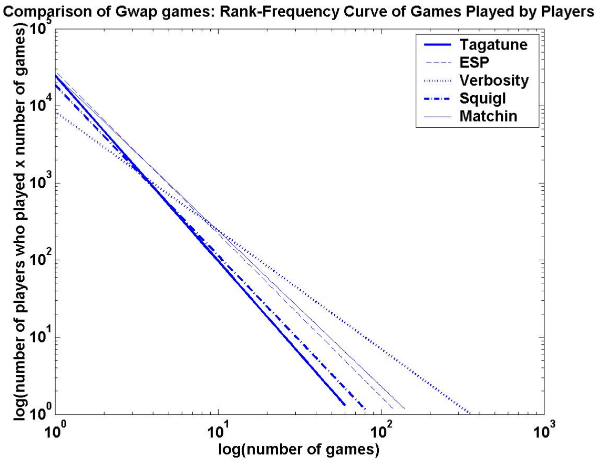


Figure 3.9: Player retention for games on GWAP.com

.

The remaining 2.64% are called missed rounds, where one or both players did not submit their vote, most likely due to a timeout of the game.

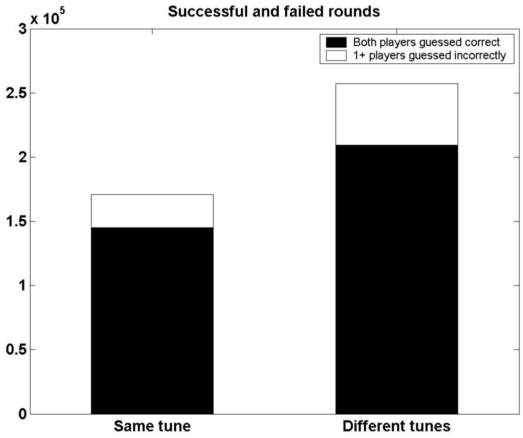


Figure 3.10: Successful versus failed rounds

.

Of the completed rounds, 80% were successful, meaning that both players guessed correctly whether they were listening to the same tune or different tunes; while 20% of the rounds were failures, where one or both players guessed incorrectly. Figure 3.10 provides a summary of these statistics. The success rate for rounds in which the tunes were the same was 85%, whereas the success rate for rounds in which the tunes were different was 81%, suggesting that it may be slightly harder to distinguish between tunes that are different.

3.2.3.5 Tag-Based Statistics

The results (see Table 3.1) show that the popularity and throughput of TagATune are superior to other human computation games for collecting music metadata. In fact, TagATune resulted in one of the largest labeled music datasets [Law 2012] publicly available to MIR researchers.

Table 3.1: Comparison of human computation games for music (some of these statistics are taken from [Mandel 2009b, Kim 2008])

	TagATune	**MajorMiner**	**The Listen Game**	**MoodSwings**
Users	14,224	490	440	100
Clips Labeled	30,237	2,300	250	1,000
Data collected	108,558 verified tags	12,000 verified tags	26,000 choices	50,000 valence-arousal labels
Unique Tags	70,908	6,400	120	Not applicable

Prior to compiling statistics on the tags, a basic level of preprocessing was performed to convert all tags into lowercase, delete leading and trailing spaces, and remove punctuation marks (such as ?, !, ., *, - and quotation marks). After preprocessing, there were a total of 512,770 tags collected, of which 108,558 were verified by at least two players and 70,908 were unique. Based on this, the average number of tags generated per minute of play is approximately four.

The 50 most frequently used tags (the “head list”) are shown in Table 3.2. There are a few observations. First, as also confirmed in other studies [Mandel 2009b, Turnbull 2007b], the most common tags used to describe music fall into the categories of genre (e.g., classical, rock, techno), instrumentation (e.g., guitar, piano, violin, drums, singing), or aspects of the music itself (e.g., fast, soft). Second, there are some tags in the “head list”–specifically ‘same,’ ‘diff,’ ‘yes,’ ‘no’–that have nothing to do with the content of the music, but instead are communication vehicles between partners in a game. Players use the words ‘same’ or ‘diff’ to signal their decision for that round to their partner. Although these communication tags are problematic, they are relatively easy to filter out since they often occur in the same formats.

A third observation is that this game generates negation tags, which are tags that describe what is not in the audio file, e.g., ’no vocals.’ This is also a consequence of communication between the partners. For example, if one player types ’singing,’ their partner might type ‘no vocals’ to indicate the difference between his or her tune and that of the partner. Other examples of negation tags include ‘no piano,’ ‘no guitar,’ ‘no drums,’ ‘not classical,’ ‘not English,’ ‘not rock,’ ‘no lyrics,’ etc. Negation tags are a unique product of TagATune and its underlying input-agreement mechanism, and are not often found in output-agreement games where communication is forbidden. Finally, even among the most frequently used tags, there are still many equivalent tags that were considered distinct due to differences in spelling, wording, and pluralization. This property of the data is useful for search, since keywords entered by users can be just as varied. However, as a dataset for training machine

Table 3.2: Head List: top 50 most frequently used tags

Tag	Count	Tag	Count
classical	37,781	no vocals	6126
guitar	30,093	soft	5,642
piano	27,718	sitar	5,642
violin	19,525	no vocal	5,285
slow	18,485	classic	5,228
strings	17,484	male	5,216
rock	17,413	singing	5,059
techno	15,627	solo	5,047
opera	14,512	vocals	5,014
drums	13,667	cello	4,966
same	12,610	loud	4,957
flute	12,149	woman	4,321
fast	11,435	pop	4,213
diff	11,046	male vocal	3,951
electronic	10,333	choir	3,576
ambient	8,733	violins	3,454
beat	7,683	new age	3,390
yes	7,352	beats	3,387
harpsichord	7,261	no voice	3,252
indian	7,255	harp	3,172
female	7,071	voice	3,080
vocal	6,964	weird	3,056
no	6,659	instrumental	2,946
synth	6,530	dance	2,896
quiet	6,167	female vocal	2,873

learning algorithms, this indicates the need for more post-processing.

In contrast to the head list, the “tail list” consists of tags that have been used very infrequently. Some of the tags are simply uncommon, e.g., ‘helicopter sound,’ ‘Halloween,’ ‘cookie monster vocals,’ ‘wedding reception music,’ etc. The tail list tags can be divided into four categories: misspelled tags, longer communication tags, compound tags (tags that contain multiple descriptors), and transcription tags (tags that transcribe lyrics). Examples of each kind are shown in Table 2.

Table 3.3: Tail List examples

Compound Tags	Transcription Tags
eastern female voice	fill me up...
long slow tones	rain on my parade
trombone and guitar	from shore to shore
light violin	the highest of sunny days
piano male voice	he’ll never love you the way
Misspelled Tags	**Communication Tags**
churhc music	pick sooner
coubtry	you have to give me info
otiental sound	you’re good too
instrumental	hello :)
ipano	yes agree

While the first two types of tail list tags are not of interest to us, compound tags can be converted into individual keywords for search, and tags which transcribe lyrics are invaluable since they support the prevalent strategy of searching for music by lyrics.

3.2.3.6 Tune-based Statistics

On average, each game serves about nine songs. After seven months of game play, there were a total of 30,237 audio clips annotated and 108,558 verified (confirmed by at least two players) tags collected. Throughout this section, the term "verified" is used to refer to tags that have high confidence (because they have been independently generated by multiple players) and "unverified" to refer to tags that have low confidence.

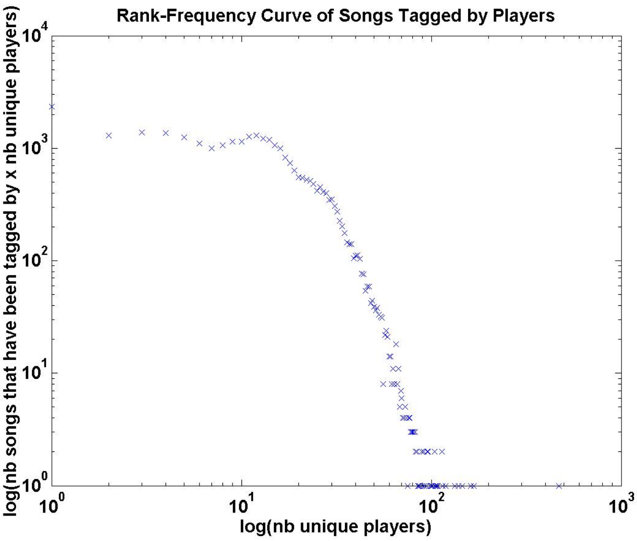


Figure 3.11: Number of songs that are tagged by x number of unique players

Figure 3.11 shows the number of the audio clips that have been tagged by x number of players. The data indicates that 92% of the audio clips have been annotated by two or more players, 61% have been annotated by ten or more players, and 26% have been annotated by 20 or more players. In order to attain a high level of confidence about the tags, an important criterion is that most songs are evaluated by multiple players. These results show that even using a simple random selection strategy for picking songs to present to players, this criterion is satisfied.

One question is whether allowing free-form text entry and open communication between partners results in tags that are accurate descriptions of the audio clips. In order to evaluate the quality of the tags, we conducted an experiment that evaluated how well the collected tags described the audio clips based on a small sample of the data.

Methodology

Twenty music clips with at least five verified tags were chosen at random, then 100 participants were solicited via Mechanical Turk (http://www.mturk.com) to answer a set of 20 questions. For each question, the participant was given a music clip and was asked to answer four sub-questions. The first two sub-questions pertained to the quality of the verified tags, i.e., tags that were confirmed by at least two players. The second two sub-questions pertained to the quality of unverified tags, i.e., tags that were entered once only for that particular audio clip. Note that the number of verified and unverified tags varies among different music clips. On average, each music clip had around 7 verified tags and 17 unverified tags. The two sub-questions for the verified tags were as follows:

1. Which of the following tags would you use to describe the piece of music to someone who could not hear it?
2. Which of the following tags have *nothing* to do with the piece of music (i.e., you don't understand why they are listed with this piece of music)?

The same two sub-questions were asked for the unverified tags; we will refer to them in order as questions 3 and 4. For each question, participants were asked to count the number of tags that would be appropriate answers and to respond by a picking a number from a combo box.

Results

We retained results from 80 of the participants who spent at least 1,000 seconds on the task, which is the time needed to listen to the entire audio clip for each question plus at least five seconds to answer each of the four sub-questions. Note that for this experiment, we did not perform any post-processing to remove the easily filterable junk words–such as 'same,' 'diff,' 'yes,' 'no'–before presenting the tags to the participants. This is because we were also interested in finding out whether there were fewer junk words among the verified tags than unverified tags.

The results of this survey are summarized in Figure 3.12. As desired, for question 1 the mean was 78.26% (s.d.=9.45), equal to roughly 5-6 out of 7 tags. This indicates that the verified tags are useful for describing the audio clip. The mean of 16.67% (s.d.=8.59) for question 2, or roughly 1 out of 7 tags, indicates that there are very few of the verified tags that do not describe the audio clip at all. This small error can be attributed mostly to the easily filterable junk words that we decided to present to the participants during this experiment (such as 'same,' 'diff,' etc.).

The results for questions 3 and 4 indicate the quality of the unverified tags. One would expect the mean percentage for question 3 to be lower than for question 1, and the mean percentage for question 4 to be higher than for question 2. This is exactly what is observed in the results. For question 3, the mean is 51.84% (s.d.=7.33), which is equivalent to 8-9 out of 17 tags, indicating that in general, a

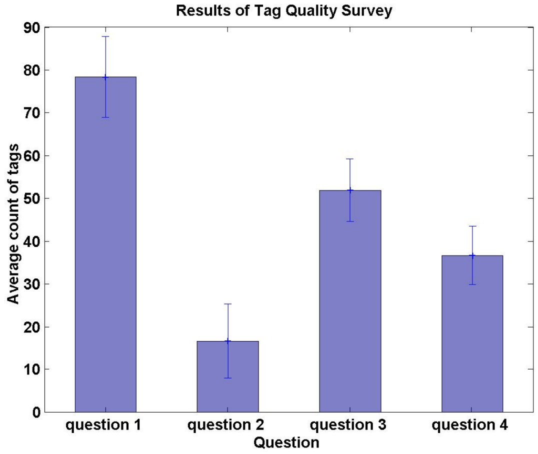


Figure 3.12: Results of questions 1-4 in tag quality survey

.

smaller proportion of the unverified tags are useful for describing an audio clip. For question 4, the mean is 36.61% (s.d.=6.8) or 6 out of 17 tags, suggesting that a greater proportion of the unverified tags have nothing to do with the content of the music than the verified tags. The difference between the percentage of good quality tags in question 1 and 3 is statis-tically significant (F(1,38)=92.74, p « 0.001), and likewise for the difference between question 2 and 4 (F(1,38)=62.96, p « 0.001). However, it is worth noting that there are usually many more unverified tags than verified tags. In some ways, the result is surprising in that a non-trivial proportion of the unverified tags actually describe the content of the music. This implies that the tail list of the collected tags is still potentially useful as data for search.

3.2.3.7 Discussion

One of the key ideas of the output-agreement mechanism first utilized in the ESP Game is that labeled data can be assigned high confidence if it is verified by multiple players, which motivated the use of agreement. We have shown that agreement is neither the only nor always the best mechanism for data extraction. In this section, we outline the major characteristics of the input-agreement mechanism and the conditions under which it is most applicable for data collection.

Multiple Levels of Verification

The input-agreement mechanism allows multiple opportunities to verify that a tag is in fact a good description for an audio clip. First, each player's descriptions are implicitly verified by their partner during the game; that is, players will only choose 'same' if they believe that their partner's descriptions are appropriate for the audio clip they themselves are listening to. Likewise, players will only choose 'different' if they believe that their partner's descriptions do not adequately describe the audio clip. In other words, the task of guessing whether the players are listening to the same or different audio clips is a good indicator of whether the tags are appropriate for the audio clips.

A second level of verification takes place offline after the data has been collected, where descriptions become official tags for the audio clip only if they are verified by greater than x players. The higher x is, the more confidence we have about the appropriateness of the descriptions for the audio clip. This utilizes the idea of agreement that is prevalent in the output-agreement games. However, the main difference here is that agreements between tags are not captured during the game, but afterwards. This is essential for collecting descriptions for data which has high description entropy–such as sounds, music, and videos–where agreement of descriptions between two partners is difficult to attain during the game, and which, in turn, may cause user dissatisfaction.

Lack of Cheating Strategies

The prevention of cheating is one of the major issues in the design of human computation games. In the ESP Game, for example, a pair of players can cheat if they settle on a strategy of typing in the same tag in order to match with each other, regardless of the content of the image. This problem is usually addressed by two countermeasures: (1) adding a delay in the player matching process so it is not guaranteed that two people who click 'play' simultaneously will be matched, and (2) giving players inputs for which the correct answers are already known.

An important property of the input-agreement mechanism is that there is no obvious strategy for cheating. While our goal is to collect tags for audio clips, the objective of the game is not to tag, but to judge from the tags entered whether the players are listening to the same audio clip. There are three basic features of the input-agreement mechanism that result in a lack of cheating strategies, as well as a lack of need for cheating: (1) neither player holds the ground truth, (2) each player must derive this ground truth from the other's descriptions, and (3) players are rewarded only if both of them obtain the ground truth. In short, by being truthful to each other, players increase their probability of obtaining the ground truth and scoring points, which as a result, generates valid descriptions for the audio clips served in the game. The pre-agreed cheating strategies that are potentially detrimental to an output-agreement game are not a problem here, because the players are allowed to communicate anyway.

Increased Complexity of Collected Tags

One of the common problems in output-agreement games is that in their efforts to match with each other, players choose to enter short, obvious, and general descriptions. This problem is alleviated, but not completely solved, by the introduction of "taboo" words. In input-agreement games, the goal is not to match on the tags, but to provide descriptions of the input data that are as detailed and accurate as possible so that the partners can guess the ground truth successfully. This allows tags to be longer and more varied. This is evident in the results obtained from the experiment presented in the previous section, showing that a non-trivial number of the longer, more complex tags in the tail list are valid descriptions of the audio clips.

Conditions of Applicability

As mentioned previously, the input-agreement mechanism can be applied to collect data about input objects with high description entropy. In fact, the TagATune game can be readily transformed to handle images, videos and text. The applicability of the input-agreement mechanism is not limited to multimedia objects. Indeed, since the launch of TagATune, two games [Law 2009a, Ma 2009] have already been developed in the domain of Web search using a modified version of the input-agreement mechanism.

3.2.4 Using TagATune for Evaluation

3.2.4.1 Motivation

An unanticipated finding was that TagATune can be used to evaluate the performance of automated music tagging algorithms. This new use of games for evaluation diverges from the conventional way to evaluate audio tagging algorithms, which involves measuring the level of agreement between the output generated by the algorithm and the ground truth set. Agreement-based metrics, e.g. accuracy, precision, F-measure and ROC curve, have been long-time workhorses of evaluation, accelerating the development of new algorithms by providing an automated way to gauge performance.

The most serious drawback to using agreement-based metrics is that ground truth sets are never fully comprehensive [Law 2008]. First, there are exponentially many sets of suitable tags for a piece of music – creating all possible sets of tags and then choosing the best set of tags as the ground truth is difficult, if not impossible. Second, tags that are appropriate for a given piece of music can simply be missing in the ground truth set because they are less salient, worded differently (e.g. *baroque* versus *17th century classical*), or that they do not facilitate the objectives of the particular annotator. For example, a last.FM user who wants to showcase his expertise on jazz music may tag the music with highly obscure and technical terms. In output-agreement games such as MajorMiner [Mandel 2009b] and the Listen Game [Turnbull 2007b], where the scoring depends on how often players' tags match with one another, players are motivated to enter (or select) tags that are common, thereby omitting tags that are rare or verbose. Furthermore, because an exhaustive set of negative tags is impossible to specify, when a tag is missing, it is impossible to know whether it is in fact inappropriate for a particular piece of music.

Agreement-based metrics also impose restrictions on the type of algorithms that can be evaluated. To be evaluated, tags generated by the algorithms must belong to the ground truth set. This means that audio tagging algorithms that are not trained on the ground truth set, e.g. those that use text corpora or knowledge bases to generate tags, cannot be evaluated using agreement-based metrics.

Finally, the *evaluation* of algorithms in Music Information Retrieval (MIR) also requires substantial human effort. MIREX [Mirex 2012] is an annual benchmarking competition for evaluating and comparing MIR algorithms, e.g., for classifying audio, measuring music similarity, detecting onsets and keys, and retrieving music via a variety of modalities. In some cases, human judges were needed to evaluate the output of the algorithms. For example, to evaluate the algorithms submitted for the "music similarity and retrieval" track, typically around 40–50 researchers are required to each invest 3–4 hours of their time to help evaluate the competing algorithms.

To be useful, tags generated by audio tagging algorithms must, from the perspective of the *end user*, accurately describe the music. However, because we do not yet fully understand the cognitive processes underlying the representation and categorization of music, it is often difficult to know what makes a tag "accurate" and what kinds of inaccuracies are tolerable. For example, it may be less disconcerting for users to receive a *folk* song when a *country* song is sought, than to receive a *sad, mellow* song when a *happy, up-beat* song is sought. Ideally, an evaluation metric should measure the quality of the algorithm by implicitly or explicitly capturing the users' differential tolerance of incorrect tags generated by the algorithms. The new evaluation metric we are proposing here has exactly this desired property.

The problems highlighted above suggest that music tagging algorithms, especially those used to facilitate retrieval, would benefit enormously from evaluation by human users. Manual evaluation is, however, often too time-consuming or costly to be feasible. Human computation is a new area of research that studies how to build systems, such as simple casual games, to collect annotations from human users. In this work, we investigate the use of a human computation game called TagATune to collect evaluations of algorithm-generated music tags. In an off-season MIREX [Downie 2008] evaluation task, we compared the performance of five audio tagging algorithms under the newly proposed metric, and present our findings.

3.2.4.2 TagATune as an Evaluation Platform

In TagATune, when a human partner is not available, a player is paired with a computer bot, which outputs tags that have been previously collected by the game for the particular music clip served in each round. This so-called *aggregate bot* serves tags that are essentially the ground truth, since they were provided by human players.

The key idea behind TagATune as an evaluation platform is that the aggregate bot can be replaced by an *algorithm bot*, which enters tags that were previously generated by an algorithm. An interesting by-product of playing against an algorithm bot is that by guessing same or different, the human player is essentially making a judgment on the appropriateness of the tags generated by the algorithm. Unlike the conventional evaluation metrics where a tag either matches or does not match a tag in the ground truth set, this evaluation method involves set-level judgments and can be applied to algorithms whose output vocabulary is ***arbitrarily different*** from that of the ground truth set.

3.2.4.3 Special TagATune Evaluation

To solicit submissions of audio tagging algorithms whose output can be used to construct the TagATune algorithm bots, a "Special TagATune Evaluation" was run off-season under the MIREX rubric. Participating algorithms were asked to provide two different types of outputs:

1. a binary classification decision as to whether each tag is relevant to each clip.
2. a real valued estimate of the 'affinity' of the clip for each tag. Larger values of the affinity score indicate that a tag is more likely to be applicable to the clip.

The Dataset

In the context of the off-season MIREX evaluation task, we trained the participating algorithms on a subset of the TagATune dataset, such that the tags they generated could be served by the algorithm bots in the game. The training and test sets comprise of 16289 and 100 music clips respectively. The test set was limited to 100 clips for both the human evaluation using TagATune and evaluation using the conventional agreement-based metrics, in order to facilitate direct comparisons of their results. Each clip is 29 seconds long, and the set of clips are associated with 6622 tracks, 517 albums and 270 artists. The dataset is split such that the clips in the training and test sets do not belong to the same artists. Genres include Classical, New Age, Electronica, Rock, Pop, World, Jazz, Blues, Metal, Punk etc. The tags used in the experiments are each associated with more than fifty clips, where each clip is associated only with tags that have been verified by more than two players independently.

Participating Algorithms

There were five submissions, which we will refer to as Mandel, Manzagol, Marsyas, Zhi and LabX[1] from this point on. A sixth algorithm we are using for comparison is called AggregateBot, which serves tags from a vocabulary pool of 146 tags collected by TagATune since deployment, 91 of which overlap with the 160 tags used for training the algorithms. The inclusion of AggregateBot demonstrates

[1] The LabX submission was identified as having a bug which negatively impacted its performance, hence, the name of the participating laboratory has been obfuscated. Since LabX essentially behaves like an algorithm that randomly assigns tags, its performance establishes a lower bound for the TagATune metric.

the utility of TagATune in evaluating algorithms that have different tag vocabulary.

Game-friendly Evaluation

An important requirement for using human computation games for evaluation is that the experiment does not significantly degrade the game experience. We describe here a few design strategies to maintain the enjoyability of the game despite the use of algorithm bots whose quality cannot be gauged ahead of time.

First, a TagATune round is randomly chosen to be used for evaluation with some small probability x. This prevents malicious attempts to artificially boost or degrade the evaluation of particular algorithms, which would be easy to do if players can recognize that they are playing against an algorithm bot. Second, while it may be acceptable to use half of the rounds in a game for evaluating good algorithms, one round may be one too many if the algorithm under evaluation always generates completely wrong tags. Since we do not know ahead of time the quality of the algorithms being evaluated, x must be small enough such that the effects of bad algorithms on the game will be minimized. Finally, using only a small portion of the game for evaluation ensures that a wide variety of music is served, which is especially important when the test set is small.

Despite the small probability of using each round for evaluation, the game experience can be ruined by an algorithm that generates tags that are contradictory (e.g. *slow* followed by *fast*, or *guitar* followed by *no guitar*) or redundant (e.g. *string*, *violins*, *violin*). Our experience shows that players are even less tolerant of a bot that appears "stupid" than of one that is wrong. Unfortunately, such errors occur quite frequently. Table 3.4 provides a summary of the number of tags generated (on average) by each algorithm for the clips in the test set, and how many of those are removed because they are contradictory or redundant.

Algorithm	Generated	Contradictory or Redundant
Mandel	36.47	16.23
Marsyas	9.03	3.47
Manzagol	2.82	0.55
Zhi	14.0	5.04
LabX	1.0	0.00

Table 3.4: Average number of tags generated by algorithms and contradictory/redundant ones among the generated tags

To alleviate this problem, we perform the following post-processing step on the output of the algorithms. First, we retain only tags that are considered relevant according to the binary outputs. Then, we rank the tags by affinity. Finally, for each tag, starting from the highest affinity, we manually remove lower affinity tags with which it is mutually exclusive. Although this reduces the number of tags available to the algorithm bots to serve in the game, we believe that this is a sensible post-processing step for any tag classification algorithms.

An alternative method of post-processing would be to first organize the "relevant" tags into categories (e.g. genre, volume, mood) and retain only the tag with the highest affinity score in each category, thereby introducing more variety in the tags to be emitted by the algorithm bots. We did not follow this approach because it may bias performance in an unpredictable fashion and favour the output of certain algorithms over others.

Evaluation Using The TagATune Metric

During an evaluation round, an algorithm is chosen to emit tags for a clip drawn from the test set. The game chooses the algorithm-clip pair in a round robin fashion but favors pairs that have been seen by the least number of unique human players. In addition, the game keeps track of which player has encountered which algorithm-clip pair, so that each evaluator for a given algorithm-clip pair is unique.

Suppose a set of algorithms $\mathcal{A} = \{a_i, \ldots, a_{|\mathcal{A}|}\}$ and a test set $\mathcal{S} = \{s_j, \ldots, s_{|\mathcal{S}|}\}$ of music clips. During each round of the game, a particular algorithm i is given a clip j from the test set and asked to generate a set of tags for that clip. To be a valid evaluation, we only use rounds where the clips given to the human player and the algorithm bot are the same. This is because if the clips are different, an algorithm can output the wrong tags for a clip and actually *help* the players guess correctly that the clips are different.

A human player's guess is denoted as $G = \{0, 1\}$ and the ground truth is denoted as $GT = \{0, 1\}$, where 0 means that the clips are the same and 1 means that the clips are different. The performance P of an algorithm i on clip j under TagATune metric is as follows:

$$P_{i,j} = \frac{1}{N} \sum_{n}^{N} \delta(G_{n,j} = GT_j) \tag{3.1}$$

where N represents the number of players who were presented with the tags generated by algorithm i on clip j, and $\delta(G_{n,j} = GT_j)$ is a Kronecker delta function which returns 1 if, for clip j, the guess from player n and the ground truth are the same, 0 otherwise. The overall score for an algorithm is averaged over the test set S:

$$P_i = \frac{1}{S} \sum_{j}^{S} P_{i,j} \tag{3.2}$$

Evaluation Using Agreement-Based Metrics

We have chosen to compute the performance of the participating algorithms using a variety of agreement-based metrics that were included in the 2008 MIREX ATC task, as a comparison against the TagATune metric. These metrics include precision, recall, F-measure, the Area Under the Receiver Operating Characteristic curve (AUC-ROC) and the accuracy of the positive and negative example sets for each tag. We omitted the “overall accuracy” metric, as it is a very biased statistics for evaluating tag classification models where there is a large negative to positive tag ratio.

As the TagATune game and metric necessarily focus on the first few tags returned by an algorithm (i.e. tags that have the highest affinity scores), we chose to also calculate the Precision-at-N ($P@N$) score for each algorithm. This additional set of statistics allows us to explore the effect of sampling the top few tags on the performance of the algorithms.

Algorithm	**TagATune metric**	Precision	Recall	**F-measure**
AggregateBot	**93.00%**	–	–	–
Mandel	**70.10%**	0.1850	**0.7313**	0.2954
Marsyas	68.60%	**0.4684**	0.4583	**0.4633**
Manzagol	67.50%	0.4574	0.1398	0.2141
Zhi	60.90%	0.2657	0.4030	0.3203
LabX	26.80%	0.03	0.0033	0.0059

Table 3.5: Evaluation statistics under the TagATune versus agreement-based metrics

Algorithm	Precision at N					Precision for ‘relevant’ tags	AUC-ROC
	3	6	9	12	15		
Mandel	0.6133	0.5083	0.4344	0.3883	0.3387	0.1850	0.8514
Marsyas	**0.7433**	**0.5900**	**0.4900**	**0.4308**	**0.3877**	**0.4684**	**0.9094**
Manzagol	0.4767	0.3833	0.3222	0.2833	0.2520	0.4574	0.7521
Zhi	0.3633	0.3383	0.3100	0.2775	0.2480	0.2657	0.6697
LabX	–	–	–	–	–	0.03	–

Table 3.6: Precision and AUC-ROC statistics collected for each algorithm

Statistical Significance

Friedman’s ANOVA [Downie 2008] is a non-parametric test that can be used to determine whether the difference in performance between algorithms is statistically significant. For each algorithm, a performance score is computed over the test set. Using the TagATune metric, this performance score is the percentage of unique players that correctly judged that the clips are the same or not using the tags emitted by the algorithm, computed using equation (1) and (2). For automated statistical evaluations, such as those performed during the MIREX ATC task, these may be the F-measure or $P@N$ for the “relevant” tags generated for each clip, or the AUC-ROC for the “affinity” scores. These scores can be viewed as a rectangular matrix, with the different tagging algorithms represented as the columns and the clips (or the tags, in the case of F-measure aggregated over each tag) forming the rows.

To avoid having variance introduced by different tags affecting the scaling and distribution of the scores, Friedman’s test replaces the performance scores with their ranks amongst the algorithms under comparison.

Friedman's ANOVA is used to determine if there exists a significant difference in performance amongst a set of algorithms. If a difference is detected, then it is common to follow up with a Tukey-Kramer Honestly Significant Difference (TK-HSD) test [Tukey 1953, Kramer 1956] to determine which pairs of algorithms are actually performing differently. This method does not suffer from the problem that multiple t-tests do where the probability of incorrectly rejecting the null hypothesis (i.e. that there is no difference in performance) increases in direct proportion to the number of pairwise comparisons conducted.

3.2.4.4 Results

Tables 3.5 and 3.6 provide summaries of the evaluation statistics collected for each algorithm under the TagATune metric as well as agreement-based metrics. Each of the summary results was computed over the 100 clips in the test set, while the statistical significance tests were computed over the results for each individual clip. The following sections detail additional statistics that were collected by the TagATune evaluation.

Algorithm Ranking

According to the TK-HSD test on the TagATune metric results, AggregateBot's performance is significantly better than all the others. A second group of equally performing algorithms consists of Mandel, Manzagol, Marsyas, and Zhi. LabX is the sole member of the worst performing group. Figure 3.13 highlights these TK-HSD performance groupings.

Several authors have speculated on the possibility of a "glass-ceiling" on the performance of current music classification and similarity estimation techniques. As identified by Aucouturier [Aucouturier 2006], many of these techniques are based on 'bag-of-frames' approaches to the comparison of the audio streams. Hence, the lack of a significant difference among the performances of the correctly functioning algorithms is not surprising.

The TK-HSD ordering of the algorithms using the F-measure scores (Table 3.5 and Figure 3.14) is different from that produced by the TagATune scores. Notably, the Marsyas algorithm significantly outperforms the other algorithms and the Zhi algorithm has improved its relative rank considerably.

These differences may be attributed to the fact that the performance of the Marsyas and Zhi algorithm is more balanced in terms of precision and recall than the Mandel algorithm (which exhibits high recall but low precision) and the Manzagol algorithm (which exhibits high precision but low recall). This conclusion is reinforced by the positive and negative accuracy scores, which demonstrate the tendency of the Mandel algorithm to over-estimate and Manzagol to under-estimate relevancy. Metrics that take into account the accuracies of all tags (e.g. F-measure) are particularly sensitive to these tendencies, while metrics that consider only the top N tags (e.g. the TagATune metric and $P@N$) are affected little by them.

These results suggest that the choice of an evaluation metric or experiment must take into account the intended application of the tagging algorithms. For example, the TagATune metric may be most suitable for evaluating retrieval algorithms that use only the highest ranked tags to compute the degree of relevance of a song to a given query. However, for applications that consider the all relevant tags regardless of affinity, e.g. unweighted tag clouds generators, the TagATune metric is not necessarily providing an accurate indication of performance, in which case the F-measure might be a better candidate.

Game Statistics

In a TagATune round, the game selects a clip from the test set and serves the tags generated by a particular algorithm for that clip. For each of the 100 clips in the test set and for each algorithm, 10 unique players were elicited (unknowingly) by the game to provide evaluation judgments. This totals to 5000 judgments, collected over a one month period, involving approximately 2272 games and 657 unique players.

One complication with using TagATune for evaluation is that players are allowed to make the decision of guessing same or different at any point during a round. This means that the number of tags reviewed by the human player varies from clip to clip, algorithm to algorithm. As a by-product of game play, players

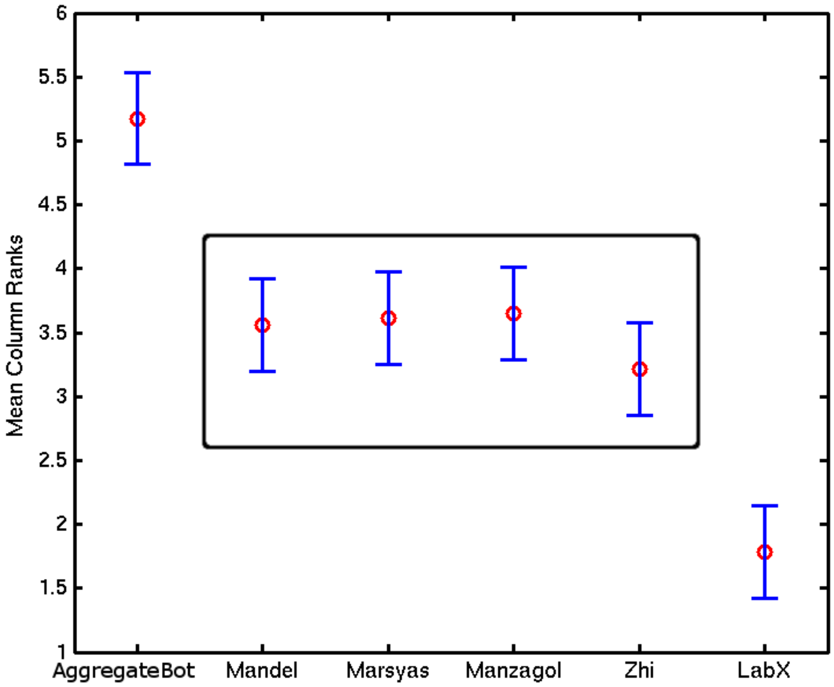


Figure 3.13: Tukey-Kramer HSD results based on the TagATune metric

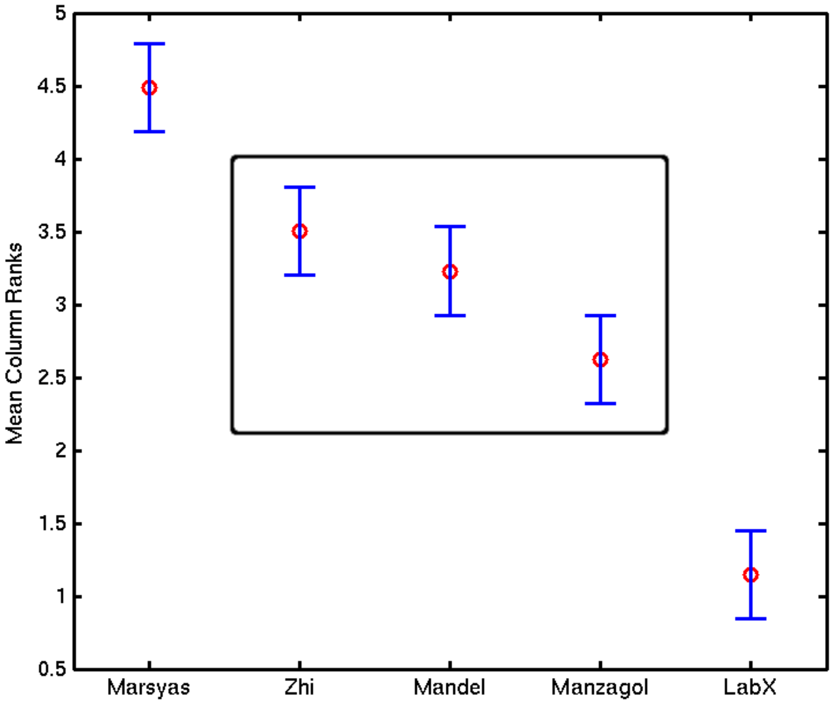


Figure 3.14: Tukey-Kramer HSD results based on the F-measure metric

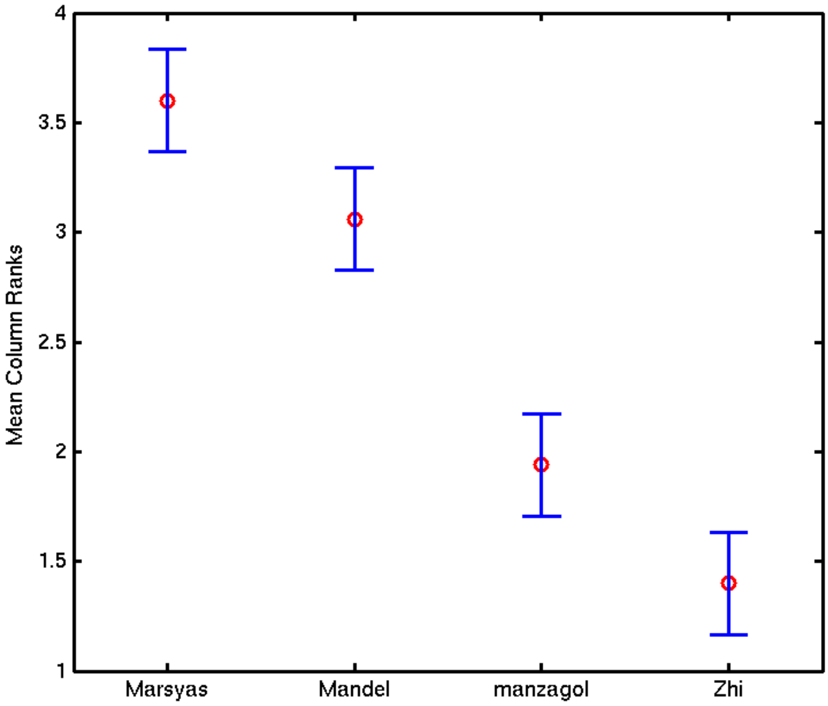


Figure 3.15: Tukey-Kramer HSD results based on the AUC-ROC metric

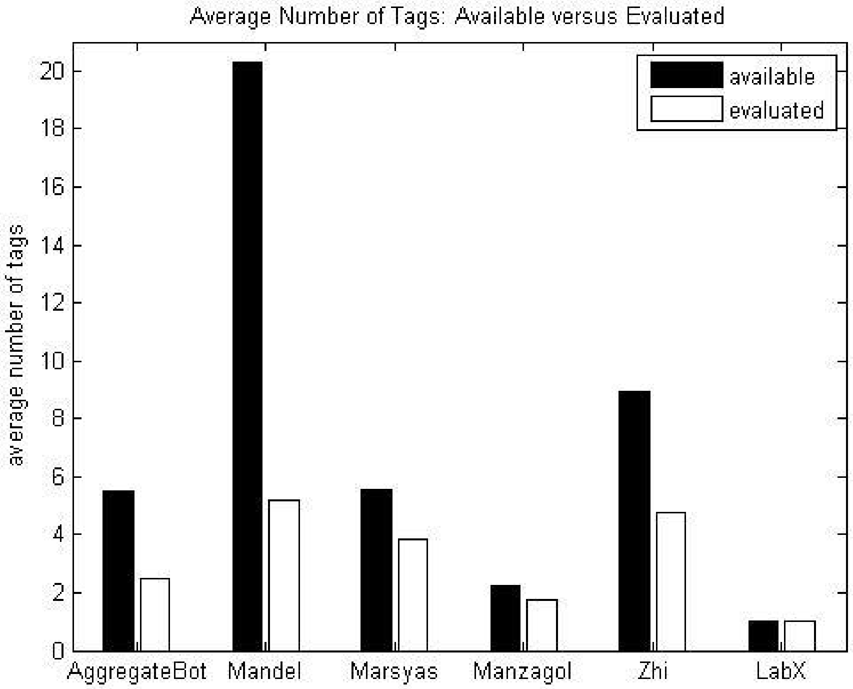


Figure 3.16: Number of tags available and reviewed by players before guessing

are motivated to guess as soon as they believe that they have enough information to guess whether the clips are the same or different. Figure 3.16, which shows that players reviewed only a small portion of the generated tags before guessing, reflects this situation.

Figure 3.17 shows the average number of tags reviewed by players and how many of the reviewed tags are actually true positive tags (according to the ground truth) in success rounds (where the human player made the correct guess) versus failed rounds (where the human player made the wrong guess). Results show that generally the number of true positive tags reviewed is greater in success rounds than in failed rounds, suggesting that players are more likely to fail at guessing when there are more top-affinity tags that are wrong. Additionally, the average number of tags reviewed before guessing is fewer in the failed rounds than in the success rounds, with the exception of Mandel, possibly due to outliers and the much greater number of tags that this algorithm returns. This suggests that players make their guesses more hastily when algorithms make mistakes.

A natural question to ask is whether one can detect from game statistics which of the reviewed tags actually caused players to guess incorrectly.

To investigate this question, we consult the game statistics for the most frequent behavior of human players in terms of the number of tags reviewed before guessing,

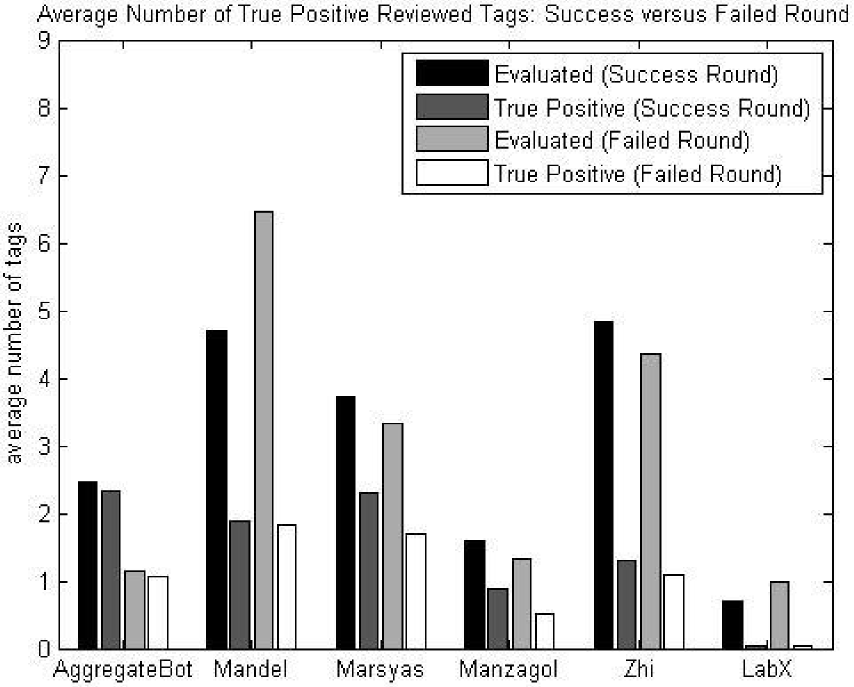


Figure 3.17: Number of overall and true positive tags evaluated in success and failed rounds

in the case when the guess is wrong. For example, we might find that most players make a wrong guess after reviewing n tags for a particular algorithm-clip pair. The hypothesis is that the last tag reviewed before guessing, i.e. the n^{th} tag, is the culprit.

System	failed round	success round
Mandel	86.15%	49.00%
Marsyas	80.49%	45.00%
Manzagol	76.92%	33.33%
Zhi	84.38%	70.10%
LabX	100.0%	95.77%

Table 3.7: Percentage of the time that the last tag displayed before guessing is wrong in a failed round versus success round

Table 3.7 shows the percentage of times that the n^{th} tag is actually wrong in failed rounds, which is above 75% for all algorithms. In contrast, the probability of the last tag being wrong is much lower in success rounds, showing that using game statistics alone, one can detect problematic tags that cause most players to make the wrong guess in the game. This trend does not hold for LabX, possibly because players were left guessing randomly due to the lack of information (since this algorithm generated only one tag per clip).

In this section, we demonstrate how TagATune is a feasible evaluation platform for collecting a large number of evaluations from human users in a timely fashion. While there are many benchmarking competitions for algorithms, little is said about the level of performance that is acceptable for real world applications. In particular, we show how aggregated data can be used as a benchmark against which algorithms are judged. Specifically, human players can correctly guess that the music are the same 93% of the times when paired against the aggregate bot, while only approximately 70% of the times when paired against an algorithm bot.

3.3 Learning from TagATune Data

In the previous section, we introduced a new human computation game called TagATune that outperforms the output-agreement mechanism for extracting music attributes from players. The remaining question is, are the data extracted by TagATune actually useful for training music tagging algorithm.

The ideal datasets used in training supervised learning algorithms should have a large and roughly equal number of examples for each class. Now consider two different versions of human computation games that collect tags for music. One solution, as prescribed in the Listen Game [Turnbull 2007c], is to randomly draw a small set of tags (e.g., "classical," "slow," "violin") from a *pre-defined* pool of 82 tags, and have game players choose the ones that describe the music clips. Another solution is to allow a pair of players complete freedom in entering any tags that come to mind. TagATune, for example, gives a pair of players two pieces of music, allow them exchange tags freely, and reward players when they can guess correctly whether the two pieces of music are the same or different. TagATune proved to be extremely fun—it has been played by tens of thousands of players, collecting over a million annotations. However, the data it collected was also extremely noisy. There are synonyms (e.g., "calm" versus "smoothing," "violins" versus "strings"), spelling mistakes, communications between players (e.g., "yes," "how are you," "different"), and compound phrases (e.g., "cookie monster vocal") that are associated with very few examples. Learning from such open vocabulary data requires the development of new methods. TagATune is a prime example of sacrificing clean data (which is the system's objective) in order to make the game more fun and attractive to players.

In this section, we explore the challenges of training machine learning algorithms using noisy data collected by human computation games. Specifically, we answer the question of whether music tagging algorithms can be trained using the noisy, open vocabulary music tags collected by TagATune. We present a new technique for classifying multimedia objects by tags, that is scalable (i.e., makes full use of the huge number of noisy labels that are freely available over the Web) and efficient (i.e., the training time remains reasonably short as the tag vocabulary grows). The main idea of our technique is to organize noisy tags into well-behaved labels using topic modeling, and learn to predict tags accurately using a mixture of topic labels. Using the TagATune [Law 2009b] dataset as a case study, we compare the tags generated by our proposed method (Topic Method) versus binary classification using tags directly as labels (Tag Method), both in terms of their relevance for each music clip, as well as their utility in facilitating the retrieval of relevant music by text. We also highlight a longstanding issue regarding the evaluation of music classifiers by ground truth set comparison, which is especially severe on open vocabulary tasks. Specifically, using the results from several Mechanical Turk studies, we show that human evaluations are essential for measuring the *true* performance of tag classifiers, which the traditional evaluation methods will consistently underestimate. In addition, tag diversity is found to be an important factor in human judgment of annotation quality not considered by most evaluation metrics or learning algorithms.

3.3.1 Motivation

In order to effectively organize and retrieve the ever growing collection of music over the Web, many automatic tag generation algorithms have been developed [Bertin-Mahieux 2008b, Hoffman 2009, Turnbull 2008b]. These so-called *music taggers* are useful for generating tags for songs that are rarely annotated by any Internet users, such as new music that just emerged on the market, or existing music belonging to lesser-known artists. Once generated, these tags can be used to support music search and recommendation on a semantic level.

In previous work, the labels used to train music taggers are considered to be devoid of errors and belonging to a small fixed vocabulary, and hence, can be directly used for training. For example, there exist music taggers using a variety of machine learning techniques, including Support Vector Machines [Li 2003, Mandel 2005], Gaussian Mixture Models [Turnbull 2007a, Turnbull 2008b], Boosting [Bertin-Mahieux 2008b], Logistic Regression [Bergstra 2006b], and other probabilistic models [Hoffman 2009]. All of these methods are trained on labels that are on the order of tens to few hundreds, as opposed to thousands to tens of thousands. For example, the Gaussian Mixture Model proposed in [Turnbull 2007a] is trained on a dataset collected from 66 paid volunteers, with 500 songs and a vocabulary size of 159 unique tags. Bertin-Mahieux et al. [Bertin-Mahieux 2008b] retained only 360 of the most popular tags from Last.FM as labels for training a artist-level tag classifier. This is in contrast to the TagATune dataset used for our experiments, which has over 30,000 clips, over 10,000 unique tags collected from tens of thousands of users.

In contrast, the tags collected by collaborative tagging websites or human computation games are noisy, i.e., they can be misspelled, redundant (due to synonyms), irrelevant to content (e.g., for organizational purpose only), and unlimited in numbers. It is difficult, from a learning perspective, to know *what classes* to learn, or determine *when* the number of examples is sufficient for training a particular class. It is also computationally inefficient to train a classifier for each tag, as the vocabulary can grow to be in the tens of thousands, or millions.

The broad problem that this work addresses is the problem of noise in datasets. Most previous work focuses on noise that is introduced when examples are misclassified into a different class [Brodley 1999, Rebbapragada 2007, Zhu 2003], and suggest a variety of methods for discarding, correcting, or re-weighting instances that are deemed incorrectly labeled, in order to improve classification accuracy.

In our work, we address a different noise problem in datasets – the over-fragmentation of the label space due to synonyms, misspelling and compound phrases. This label noise problem is readily found in the tags produced by collaborative tagging websites (such as last.FM) [Lamere 2008b] and human computation games such as TagATune [Law 2009b], where an *open vocabulary* is allowed.

Source	Type	Example
last.FM	content irrelevant	albums I own, favorites, awesome
	synonyms	deutsch, german
	misspelling	harpsicord (harpsichord)
	compound	eclectic celtic, political hip-hop
TagATune	content irrelevant	hello, you're good too, yes agree
	synonyms	choir, choral, chorus, singing
	misspelling	chello (cello), ipano (piano) vioin (violin)
	compound	country techno, guitar plucking

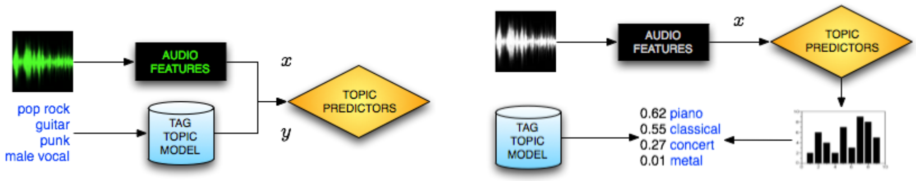
Table 3.8: Examples of Noisy Tags.

Table 3.8 shows examples of noisy tags from last.FM and TagATune by types. First, some tags are irrelevant to the audio characteristics of the music, and serve only the purpose of organization (e.g., "albums I own"), expression of opinions (e.g., "awesome"), or communication with the partner, in the case of games (e.g., "hello"). The second, and likely the most common, type of noise are synonyms and misspellings, which render music that should be in the same class to belong to different classes. Finally, a large portion of the tags are compound phrases with multiple descriptors. These tags tend to be highly specific, but are associated with very few music clips. When used as labels to train a music tagger, compound tags result in classes that contain very few positive examples.

Some recent work focuses on mitigating the problem of noisy tags from collaborative tagging websites, by learning the distinction between content relevant versus content irrelevant tags [Iwata 2009], or by discovering higher level concepts using co-occurrence statistics in the tags [Laurier 2009, Levy 2007]. However, none of these work explored the use of these higher-level concepts as labels in training annotation and retrieval algorithms.

3.3.2 Problem Formulation

This section presents the music annotation and retrieval problem formally. All vector quantities are denoted in **bold**. In both problems, we are given as training data a set of N music clips $\mathcal{C} = \{c_1, \ldots, c_N\}$ each of which has been annotated by humans using tags $\mathcal{T} = \{t_1, \ldots, t_V\}$ from a vocabulary of size V. Each music clip $c_i = (\vec{a_i}, \vec{r_i})$ is represented as a tuple, where $\vec{a_i} = \mathbb{Z}^V$ is a the *ground truth tag* vector containing the frequency of each tag in $\mathcal{T}$ that has been used to annotate the music clip by humans, and $\vec{r_i} = \mathbb{R}^M$ is a vector of M real-valued acoustic features, which describes the characteristics of the audio signal itself.



(a) Training (b) Inference

Figure 3.18: The training and inference phase of the proposed model

The goal of music annotation is to learn a function $\hat{f} : R \times T \rightarrow \mathbb{R}$, which maps the acoustic features of each music clip to a set of scores that indicate the relevance of each tag for that clip. Having learned this function, music clips can be retrieved for a search query q by rank ordering the distances between the query vector (which has value 1 at position j if the tag t_j is present in the search query, 0 otherwise) and the tag probability vector for each clip. Following [Turnbull 2008b], these distances are measured using KL divergence, which is a common measure of distance between two distributions. Note that the query vector is a valid multinomial distribution (i.e. sums to 1) for one-word queries, which are what we used to evaluate retrieval performance in this work.

3.3.3 Proposed Algorithm

As mention previously, most prior works train music taggers using the ground truth tags directly as labels. This training approach becomes infeasible when ground truth tags are collected by applications, such as collaborative tagging websites or human computation games, that do not enforce a controlled vocabulary. In this work, we propose an new method of generating tags, by first learning a mapping from audio features to a small set of topic labels that can cover all tags in the vocabulary, then using these high-level labels to recover the tags that are the most relevant for any music clip.

The inspiration of our approach comes from the work by Palatucci et al [Palatucci 2009] on zero-shot learning, where the problem is to learn a classifier to predict a huge number of labels, many of which can be missing from the training set. The particular application they are interested in, is predicting the word that a person is thinking about (e.g., dog) from the fMRI image of that person's brain. To train such a classifier using supervised learning, one would need to create a dataset containing multiple fMRI images corresponding to each word in the English language, which would be too costly. Instead, the authors advocate an alternative method of mapping image features to a set of semantic codes that can cover all words in the English language (e.g., a boolean vector indicating the answers to questions such as "Does it breathe under water?", "Is it slow moving?", "Is it furry?", "Is it carnivorous?" etc). Given a new fMRI image, the classifier can predict the semantic code of that image, then find the word in the knowledge base whose semantic code is closest to the prediction [Palatucci 2009].

In this section, we will describe in detail the training and inference phase of our proposed method, as depicted in Figure 3.18.

3.3.3.1 Training Phase

Our training phase (Figure 3.18(a)) is a two stage process. In the first stage, we induce a topic model using the ground truth tags associated with each music clip in the training set. This topic model allows us to infer the topic distribution of each music clip in the training set, and use these inferred topic distributions as new labels. The second stage involves training a classifier to predict topic distributions from audio features.

Stage 1: Topic Modeling using LDA

A topic model [Blei 2003, Steyvers 2007] is a hierarchical probabilistic model that describes the process for generating the constituents of an entity (e.g., words of an article [Erosheva 2004], musical notes in a score [Hu 2009], or pixels in an image) from a set of latent topics. In the first stage of the training phase, our goal is to drastically reduce the size of the label space, from thousands of *tag* labels to tens of *topic* labels, by learning a set of topics over the ground truth music tags that were collected by TagATune.

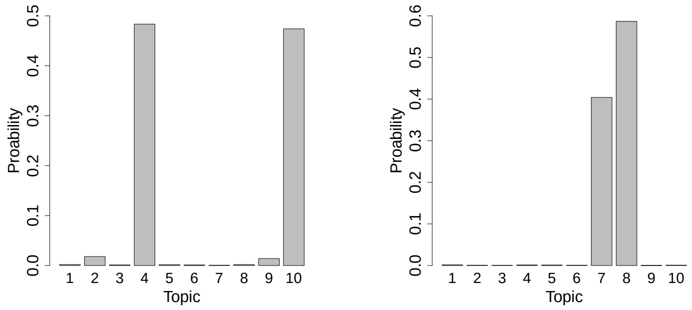
In our topic model, each topic is a distribution over music tags, and each music clip is associated with a set of topics with different probabilities. Figure 3.19(a) shows an example of a topic model (with 10 topics) learned over the music tags collected by TagATune. Figure 3.19(b) and Figure 3.19(c) show the topic distributions for two very distinct music clips and the ground truth tags associated with them (in the caption). The music clip represented by Figure 3.19(b) is associated with topic 4 (the “classical violin” topic) and topic 10 (the “female opera singer” topic), and the music clip represented by Figure 3.19(c) is associated with topic 7 (the “flute” topic) and topic 8 (the “quiet ambient music” topic).

In this work, we adopt a widely used method in topic modeling called the Latent Dirichlet Allocation (LDA) [Blei 2003], as depicted in Figure 3.20. Given N music clips, V unique tags, and K topics, LDA is a probabilistic *latent variable model*, where the observed variables (shaded in grey) are $a_{i,j}$, the ground truth tags associated with music clip c_i, and the hidden variables to be inferred (circled in **bold**) are: (i) θ_i, the topic distribution for each music clip c_i, (ii) Ψ_k, the probability of each ground truth tag a_j in topic k, and (iii) $Z_{i,j}$ the topic responsible for generating the ground truth tag $a_{i,j}$ for music clip c_i, where $i = 1, \ldots, N$ and $j = 1, \ldots, V$, and $k = 1, \ldots, K$.

The central innovation in LDA, over other topic model formulations such as Probabilistic Latent Semantic Indexing (pLSI) [Hofmann 1999], is the use of a Dirichlet prior on the topic distribution θ_i (with hyperparameters $\alpha = \alpha_1 = \cdots = \alpha_K$) and on the tag distribution Ψ_k for each topic (with hyperparameter β). These

1	electronic beat fast drums synth dance beats jazz
2	male choir man vocal male_vocal vocals choral singing
3	indian drums sitar eastern drum tribal oriental middle_eastern
4	classical violin strings cello violins classic slow orchestra
5	guitar slow strings classical country harp solo soft
6	classical harpsichord fast solo strings harpsicord classic harp
7	flute classical flutes slow oboe classic clarinet wind
8	ambient slow quiet synth new_age soft electronic weird
9	rock guitar loud metal drums hard_rock male fast
10	opera female woman vocal female_vocal singing female_voice vocals

(a) Topic Model



(b) woman, classical, classsical, opera, male, violen, violin, voice, singing, strings, italian

(c) chimes, new age, spooky, flute, quiet, whsitling, whistle, fluety, ambient, chime, snare, soft, high pitch, bells

Figure 3.19: An example of a topic model learned over music tags, and the representation of two music clips by topic distribution.

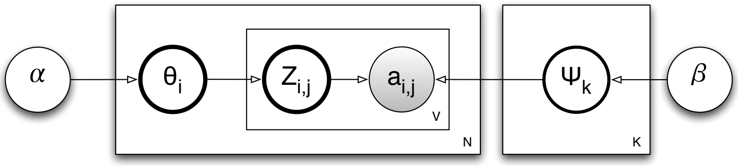


Figure 3.20: Latent Dirichlet Allocation Model.

hyperparameters are fixed.

Together, LDA specifies a joint distribution over observed and hidden variables. The inference problem, then, is to learn the parameters of the posterior probability distribution of the hidden variables (θ_i, Ψ_k, $Z_{i,j}$) conditioned on the observed data ($a_{i,j}$) and the hyperparameters (α, β). Because it is intractable to learn this posterior distribution exactly, approximate methods (e.g., Mean Field Variational

Inference [Blei 2009], Gibbs Sampling [Steyvers 2007]) have been used to solve LDA. The particular implementation used in this work is provided by the Mallet toolkit [McCallum 2002], which uses the Gibbs Sampling method specified in Steyvers et al [Steyvers 2007].

LDA provides an interesting generative story about how players of TagATune might have generated the tags for the music clips they are listening to. According to the model, each player of TagATune would have a topic structure in mind when describing music. Given a music clip, the player first selects a topic according to the topic distribution for that clip, then generates a tag according to the tag distribution of the chosen topic. Under this interpretation, our goal in building a topic model over tags is to discover the topic structure that the players used to generate tags for music, so that we can leverage a similar topic structure to automatically tag new music.

Stage 2: Topic Distribution Classification by Maximum Entropy

The topic model derived in stage 1 of the training phase can be used to assign a *ground truth topic distribution* to each music clip. In the second stage, our goal is to learn a function g that maps audio features to topic distributions, using the ground truth topic distributions as labels for training. Our classifier of choice is Maxent (maximum entropy classifier) [Csiszar 1996], which has been used extensively in text classification [Berger 1996, Nigam 1999], but to our knowledge, rarely for music tagging. The particular implementation we adopted is from the Mallet toolkit [McCallum 2002], which uses Limited Memory BFGS [Nocedal 1995] to maximize the likelihood of the parameters, and a slight modification of the optimization procedure provided by Yao et al [Yao 2009] to enable the use of topic distributions, instead of a single topic, as labels for training the classifier.

The training phase of our method is a currently a two-step process: we first learn a topic model over tags, then learn a mapping from audio features to topic distributions. In the future, we may investigate a training procedure that combines the two steps into one. For example, there has been recent work on topic models that are learned from not only text, but other metadata associated with the documents, such as sLDA [Blei 2007] and DMR [Mimno 2008]. Another class of methods to investigate are semi-supervised techniques for performing factorization and classification simultaneously, such as Support Vector Decomposition Machine (SVDM) [Pereira 2006] or Collective Matrix Factorization [Singh 2008].

3.3.3.2 Inference Phase

Figure 3.18(b) depicts the process of generating tags for new music clips. For an unseen music clip c' and given only its audio features, we can use the function g learned in stage 2 of the training phase to infer the topic distribution of that clip. Given this predicted topic distribution, each tag can be given a relevance score for the music clip c', by multiplying the probability of that tag in each topic and the probability of that topic in c', summing over all topics, i.e.

$$p(t_j|r_i) = \sum_{k}^{K} p(t_j|z=k) \cdot p(z=k|r_i)$$

where $j = 1, \ldots, V$, $i = 1, \ldots, N$ and $k = 1, \ldots, K$, In reality, there are many different ways to generate tags from a topic model. For example, one can add a restriction that says that the generated tags can only come from the top Q topics, where $Q << K$. In future work, we may experiment with different inference schemes, and compare their effectiveness in generating relevant tags for music.

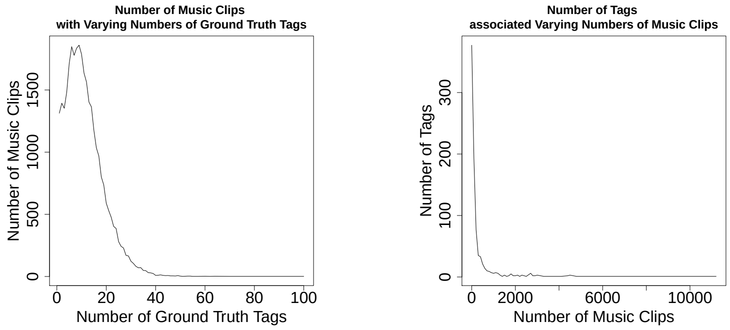
3.3.4 Experiments

Our goal is to compare our proposed method (Topic Method) against the methods of generating tags using binary classification (Tag Method) or at random (Random Method), using 5-fold cross validation. The experiments are guided by five central questions:

Feasibility	Given a set of noisy music tags, is it possible to learn a reduced representation of the tag space that is (i) semantically meaningful, and (ii) predictable by content-based features (e.g., timbre, rhythm etc) of the music?
Annotation	How accurate are the generated tags?
Retrieval	How well do the generated tags facilitate music retrieval?
Efficiency	How do the training times compare between methods?
Evaluation	To what extent are the evaluations a reflection of the true performance of the tag classifiers?

3.3.4.1 Dataset

The dataset consists of music features and tags collected by TagATune. Figure 3.21 shows the characteristics of the TagATune dataset, in terms of how many ground truth tags each music clip has, and how many music clips are available to each tag as training examples. Figure 3.21(a) is a rank frequency plot showing the number of music clips (y-axis) that have a certain number of ground truth tags (x-axis). The plot reveals that a majority of the music clips (> 1500) have under 10 ground truth tags, with around 1300 music clips with only 1 or 2 ground truth tags, and very few music clips that have a large number (e.g., > 100) of ground truth tags. This disparity in the number of ground truth tags creates a problem in our evaluation – many of the generated tags will not be found amongst the ground truth tags, and therefore will be considered incorrect when they are in fact correct. Figure 3.21(b) is a rank frequency plot showing the number of tags that have a certain number of music clips available to them as training examples. The plot shows that the vast majority of the tags have few music clips to use as training examples, while a small number of tags are endowed with a large number of examples. This highlights the aforementioned sparsity problem that emerges when tags are used directly as labels, a problem that is addressed by our proposed method.



(a) Number of music clips that have X number of ground truth tags

(b) Number of tags that are associated X number of music clips

Figure 3.21: Characteristics of the TagATune Dataset

We did a small amount of pre-processing on a subset of the data collected by TagATune until April 2009, tokenizing tags, removing punctuation and four extremely common tags that are not related to the content of the music, i.e. “yes”, “no”, “same”, “diff”. These tags are natural consequences of the game, since players communicate with each other in other ways beyond just describing the music [Law 2009b], such as saying “yes” or “no” to confirm whether the partner’s tags also describe one’s own music clip, or “same” or “diff” to notify the partner of the player’s current guess of whether the music is the same or different.

We also eliminated tags that have fewer than 20 music clips available as training examples, in order to conduct a comparison against the Tag Method, which requires sufficient amount of training examples for each binary classification task. This reduces the number of music clips from 31867 to 31251, and the total number of ground truth tags from 949,138 to 699,440, and the number of *unique* ground truth tags from 14506 to 854. For the purpose of comparison, this reduced set of ground truth tags is used in both the Topic Method and the Tag Method. Note that we are throwing away a substantial amount of tag data when we require that each tag be associated with a minimum number of examples. A motivation for using topic models to generate tags is that we do not need to throw away any tags at all. Rare tags, i.e. tags that are associated with only one or two music clips, can still be grouped into a topic, and used in the annotation and retrieval process.

Each of the 31251 music clips is 29 seconds in duration, and is represented by a set of ground truth tags collected via TagATune, as well as a set of content-based (spectral and temporal) features extracted using the technique described in [Mandel 2009a]. Spectral features consist of summary statistics (mean and covariance) of a clip's Mel-Frequency Cepstral Coefficients (MFCC), which describe the power spectrum of an audio signal on a scale composed of frequencies that are meaningful to human hearing. Temporal features describe the total magnitude of different frequency levels over time. The detail of this feature extraction scheme is available in [Mandel 2009a].

3.3.4.2 Experiment 1: Feasbility

Table 3.9 shows the top 10 words of each topic learned by LDA using the tags collected via TagATune with the number of topics fixed at 10, 20 and 30. In general, the topics are able to capture meaningful grouping of tags, e.g., synonyms (e.g., {“choir”, “choral”, “chorus”}, or {“male”, “man”, “male_vocal”, “male_voice”}), misspellings (e.g., {“harpsichord”, “harpsicord”} or {“cello”, “chello”}), or associations (e.g., {“indian”, “drums”, “sitar”, “eastern”, “tribal”, “oriental”} or {“rock”, “guitar”, “loud”, “metal”}). As we increase the number of topics, there emerge new topics that are not captured by topic models with fewer number of topics. For example, in the topic model with 20 topics, topic 3 (which describes soft classical music), topic 13 (which describes jazz), topic 17 (which describes rap, hip-hop and reggae) are new topics that are not evident in the topic model with 10 topics. We also observe some repetition (or refinement) of topics as the number of topic increases (e.g., topics 8, 25 and 27 in the 30-topic model all describe female vocal music, but are slightly different in terms of genre).

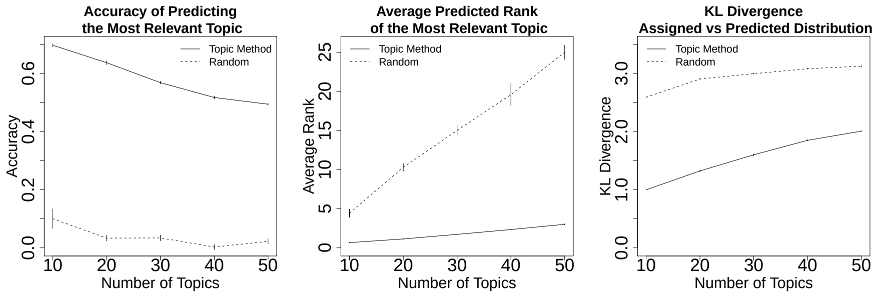
It is difficult to know exactly how many topics can succinctly capture the concepts underlying the music in our dataset. For all our experiments, we empirically tested how well topic distribution and the best topic can be predicted using audio features, fixing the number of topics at 10, 20, 30, 40, and 50 topics. Figure 3.22 summarizes the results. We evaluated performance using several metrics, including accuracy and average rank of the most relevant topic, as well as the KL divergence between the ground truth and the predicted topic distribution. Although we see a degration of performance as the number of topics increases, all models (under the accuracy, average rank, KL divergence metrics) significantly outperform the random baseline, which uses random distributions as labels for training. Moreover, even with 50 topics, the average rank of the most relevant topic is still around 3, which suggests that the classifier is capable of predicting the most relevant topic well. This is crucial, as the most appropriate tags for a music clip are likely to be found in the most relevant topics for that clip.

	10 Topics
1	electronic beat fast drums synth dance beats jazz electro modern
2	male choir man vocal male_vocal vocals choral singing male_voice pop
3	indian drums sitar eastern drum tribal oriental middle_eastern foreign fast
4	classical violin strings cello violins classic slow orchestra string solo
5	guitar slow strings classical country harp solo soft quiet acoustic
6	classical harpsichord fast solo strings harpsicord classic harp baroque organ
7	flute classical flutes slow oboe classic clarinet wind pipe soft
8	ambient slow quiet synth new_age soft electronic weird dark low
9	rock guitar loud metal drums hard_rock male fast heavy male_vocal
10	opera female woman vocal female_vocal singing female_voice vocals female_vocals voice

	20 Topics
1	indian sitar eastern oriental strings middle_eastern foreign guitar arabic india
2	flute classical flutes oboe slow classic pipe wind woodwind horn
3	**slow quiet soft classical solo silence low calm silent very_quiet**
4	male male_vocal man vocal male_voice pop vocals singing male_vocals guitar
5	cello violin classical strings solo slow classic string violins viola
6	opera female woman classical vocal singing female_opera female_vocal female_voice operatic
7	female woman vocal female_vocal singing female_voice vocals female_vocals pop voice
8	guitar country blues folk irish banjo fiddle celtic harmonica fast
9	guitar slow classical strings harp solo classical_guitar soft acoustic spanish
10	electronic synth beat electro ambient weird new_age drums electric slow
11	drums drum beat beats tribal percussion indian fast jungle bongos
12	fast beat electronic dance drums beats synth electro trance loud
13	**jazz jazzy drums sax bass funky guitar funk trumpet clapping**
14	ambient slow synth new_age electronic weird quiet soft dark drone
15	classical violin strings violins classic orchestra slow string fast cello
16	harpsichord classical harpsicord strings baroque harp classic fast medieval harps
17	**rap talking hip_hop voice reggae male male_voice man speaking voices**
18	classical fast solo organ classic slow soft quick upbeat light
19	choir choral opera chant chorus vocal vocals singing voices chanting
20	**rock guitar loud metal hard_rock drums fast heavy electric_guitar heavy_metal**

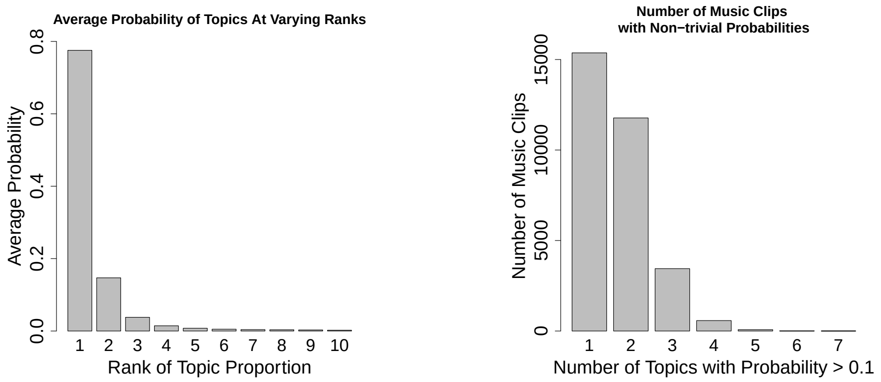
	30 Topics
1	choir choral opera chant chorus vocal male chanting vocals singing
2	classical solo classic oboe fast slow clarinet horns soft flute
3	rap organ talking hip_hop voice speaking man male_voice male man_talking
4	rock metal loud guitar hard_rock heavy fast heavy_metal male punk
5	guitar classical slow strings solo classical_guitar acoustic soft harp spanish
6	cello violin classical strings solo slow classic string violins chello
7	violin classical strings violins classic slow cello string orchestra baroque
8*	female woman female_vocal vocal female_voice pop singing female_vocals vocals voice
9	bells chimes bell whistling xylophone whistle chime weird high_pitch gong
10	ambient slow synth new_age electronic soft spacey instrumental quiet airy
11	rock guitar drums loud electric_guitar fast pop guitars electric bass
12	slow soft quiet solo classical sad calm mellow very_slow low
13	water birds ambient rain nature ocean waves new_age wind slow
14	irish violin fiddle celtic folk strings clapping medieval country violins
15	electronic synth beat electro weird electric drums ambient modern fast
16	indian sitar eastern middle_eastern oriental strings arabic guitar india foreign
17	drums drum beat beats tribal percussion indian fast jungle bongos
18	classical strings violin orchestra violins classic orchestral string baroque fast
19	quiet slow soft classical silence low very_quiet silent calm solo
20	flute classical flutes slow wind woodwind classic soft wind_instrument violin
21	guitar country blues banjo folk harmonica bluegrass acoustic twangy fast
22	male man male_vocal vocal male_voice pop singing vocals male_vocals voice
23	jazz jazzy drums sax funky funk bass guitar trumpet reggae
24	harp strings guitar dulcimer classical sitar slow string oriental plucking
25*	vocal vocals singing foreign female voices women woman voice choir
26	fast loud upbeat quick fast_paced very_fast happy fast_tempo fast_beat faster
27*	opera female woman vocal classical singing female_opera female_voice female_vocal operatic
28	ambient slow dark weird drone low quiet synth electronic eerie
29	harpsichord classical harpsicord baroque strings classic harp medieval harps guitar
30	beat fast electronic dance drums beats synth electro trance upbeat

Table 3.9: Topic Model with 10, 20, and 30 topics. The topics in bold in the 20-topic model are examples of new topics that emerge when the number of topics is increased from 10 to 20. The topics marked by * in the 30-topic model are examples of repeated or refined topics that emerge as the number of topics is increased.



(a) Accuracy (b) Average Rank (c) KL Divergence

Figure 3.22: Results showing how well topic distributions or the best topic can be predicted from audio features. The metrics include accuracy and average rank of the most relevant topic, and KL divergence between the assigned and predicted topic distribution.



(a) Average probability of topics at different rank (i.e. rank 1 = most relevant topic, rank 10 = least relevant topic)

(b) Number of music clips whose topic distribution contains X number of topics with non-trivial (> 0.1) probability

Figure 3.23: Most music clips are assigned only 1 or 2 topics with non-trivial probabilities.

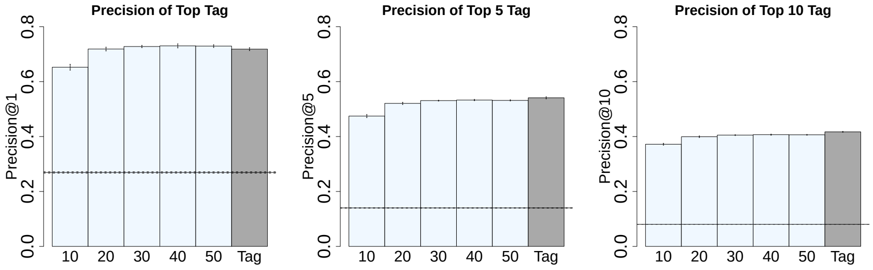
We also experimented with using the most relevant topic as the label to train the maximum entropy classifier, and observe that it produced the same results as using the topic distribution as a label for training. There are two possible explanations. First, Yao et al [Yao 2009] reported a similar observation, that the "output of the topic proportion classifier is often overly concentrated on the single largest topic". Therefore, this phenomenon can be an artifact of the particular classifier and optimization method we used. Second, we observe that for most music clips in the TagATune dataset, the topic model assigns very high probabilities to only a few topics, and low probabilities for all other topics. Figure 3.23(a) shows the probability of topics at different rank (rank 1= most relevant topic, rank 10 = least relevant topic), averaged over all music clips. It reveals that the most relevant topic has average probability of approximately 0.7, followed by the second ranking topic with probability < 0.2, and the third ranking topic with probability < 0.05, and the rest of the topics with very small probabilities. Figure 3.23(b) is a rank frequency plot showing the number of music clips whose topic distribution have X number of topics with non-trivial (> 0.1) probability. It is evident that for the majority of music clips in the TagATune dataset, their topic distributions contain only 1 or 2 topics with non-trivial probabilities.

3.3.4.3 Experiment 2: Annotation Performance

Following [Hoffman 2009], we evaluate the accuracy of the top 10 tags for each music clip, under three different metrics: per-clip metric, per-tag metric and omission-penalizing per-tag metric.

Per-Clip Metric

The per-clip precision@N metric measures the proportion of correct tags (according to agreement with the ground truth set) amongst the N tags that have the highest inferred probabilities for each clip, averaged over all the clips in the test set. The results are presented in Figure 3.24.



(a) Precision@1 (b) Precision@5 (c) Precision@10

Figure 3.24: Per-clip Metrics. The light-colored bars represent Topic Method with 10, 20, 30, 40 and 50 topics. The dark-colored bar represents the Tag Method. The horizontal line represent the random baseline, and the dotted lines represent its standard deviation.

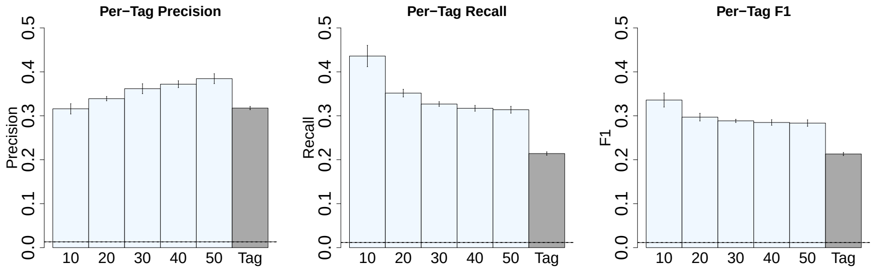
Topic Model (using 50 topics) and the Tag Method are almost indistinguishable under this metric.

Per-Tag Metric

Alternatively, we can evaluate annotation performance by computing the precision, recall and F-1 measures for each tag, averaged over all the tags that are outputted by the algorithm (i.e. if the music tagger does not output a tag, the scores for that tag are simply ignored). Specifically, given a tag t, its precision P_t, recall R_t and F-1 measure F_t can be computed as follows:

$$P_t = \frac{c_t}{a_t} \qquad R_t = \frac{c_t}{g_t} \qquad F_t = 2 \cdot \frac{P_t \cdot R_t}{P_t + R_t}$$

where g_t is the number of music clips that has the tag t in their ground truth sets, a_t is the number of clips that are annotated with the tag t by the tagger, and c_t is the number of clips that has been *correctly* annotated with the tag t by the tagger, according to the ground truth set. The overall per-tag precision, recall and F-1 scores for a test set are P_t, R_t and F_t for each tag t, averaged over all tags in the vocabulary.



(a) Per-Tag Precision (b) Per-Tag Recall (c) Per-Tag F-1

Figure 3.25: Per-tag Metrics. The light-colored bars represent Topic Method with 10, 20, 30, 40 and 50 topics. The dark-colored bar represents the Tag Method. The horizontal line represent the random baseline, and the dotted lines represent its standard deviation.

Results (in Figure 3.25) show that the Topic Method significantly outperforms the Tag Method under this set of metrics.

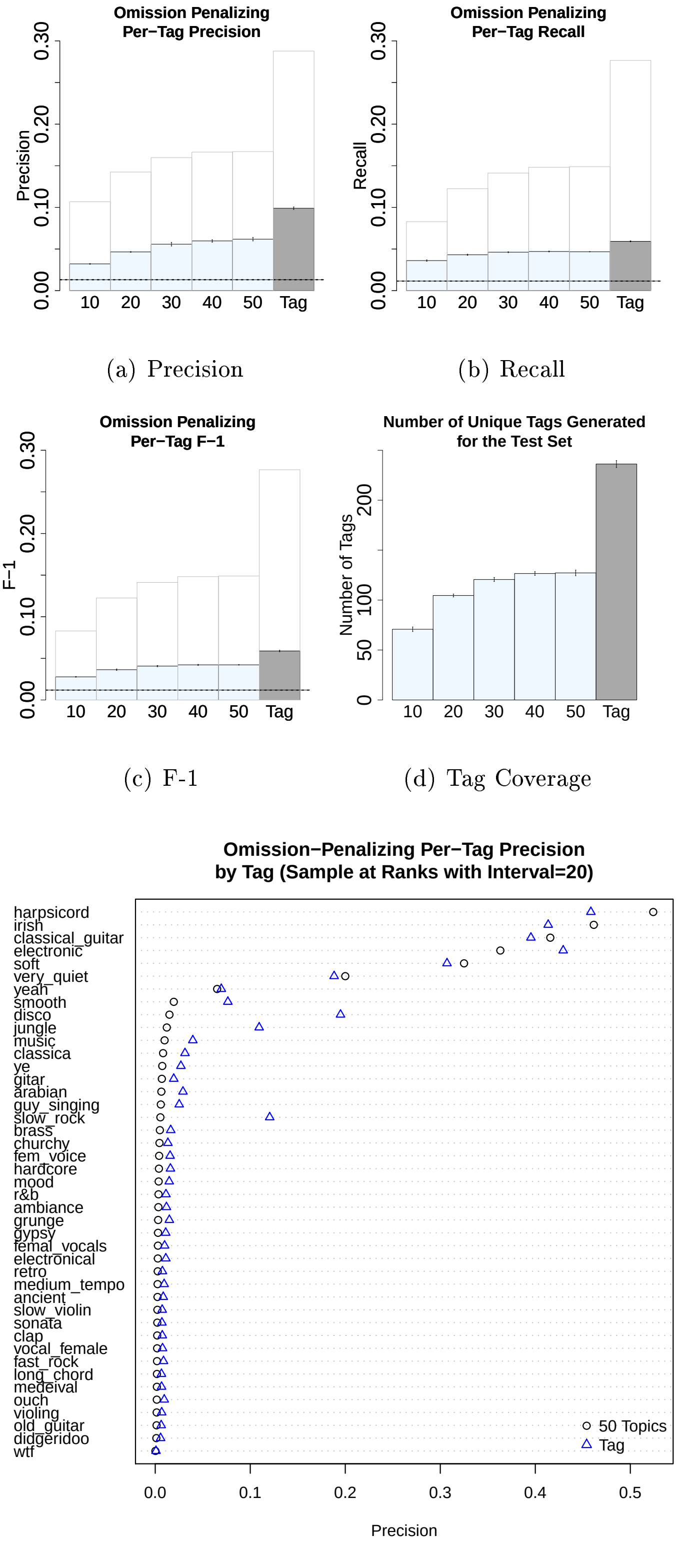
Per-Tag Metric (Omission Penalizing)

Although informative, two of the metrics – per-clip precision@N and per-tag precision – are problematic in that a system can output the most common tags, leaving out the rare ones, and still perform reasonably well under these metric [Turnbull 2008b]. In response to this criticism, several previous work [Bertin-Mahieux 2008b, Hoffman 2009, Turnbull 2008b] has adopted a set of *per-tag* metrics that penalizes algorithms for omitting tags that could have been used to annotate music clips in the test set.

Following [Hoffman 2009, Turnbull 2008b], the omission-penalizing per-tag precision and recall can be computed as follows:

$$P_t = \begin{cases} \frac{c_t}{a_t} & \text{if present} \\ E_t & \text{if omitted} \end{cases} \qquad R_t = \begin{cases} \frac{c_t}{g_t} & \text{if present} \\ 0 & \text{if omitted} \end{cases}$$

where E_t is the empirical frequency of the tag t in the test set. This specification penalizes classifiers that leave out tags, especially ones that are rare. Note that



(a) Precision

(b) Recall

(c) F-1

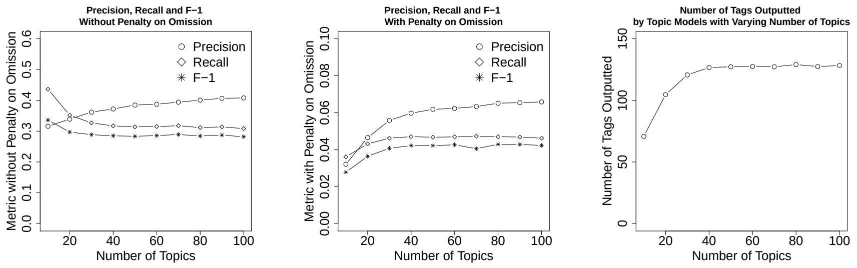
(d) Tag Coverage

(e) Precision by Tag

Figure 3.26: Omission-Penalizing Per-tag Metrics. The light-colored bars represent Topic Method with 10, 20, 30, 40 and 50 topics. The dark-colored bar represents the Tag Method. The horizontal line represent the random baseline, and the dotted lines represent its standard deviation. Figure (e) shows the precision of individual tags at rank 1, 21, 41, $\cdots$, 854 etc. It is evident that the Topic Method loses in precision by failing to output many of the rarer tags.

these metrics are upper bounded by a quantity that depends on the number of tags outputted by the algorithm. This quantity can be computed empirically by setting the precision and recall to 1 when the tag are present, and E_t and 0 when a tag is omitted.

Results (Figure 3.26 (a)–(d)) shows that for the Topic Method, performance increases with more topics, but reaches a platform as the number of topics approaches 50. We investigated additional models with 60, 70, 80, 90, and 100 topics, and found that this plateau persists in these models. In particular, Figure 3.27(a) shows that under the per-tag metric, precision keeps increasing when we increase the number of topics, but recall hits a plateau. The same performance plateau is observed under the omission-penalizing per-tag metric (Figure 3.27(b)).



(a) Per-Tag Metric

(b) Per-Tag Metric (Omission Penalizing)

(c) Tag Coverage

Figure 3.27: How performance varies as the number of topics increases.

The performance plateau can be attributed to the fact that the number of tags outputted by the topic models plateau at around 127 (Figure 3.27(c)). This is a somewhat expected, and problematic, behavior of the Topic Method, where common tags (e.g., classical) tend to be ranked higher in any given topic, and therefore, are more likely to be generated. The plateau also explains why the Tag Method outperforms the Topic Method under this metric – it generated roughly twice the number of unique tags (Figure 3.26(d)).

Figure 3.28, 3.29 and 3.30 shows the detailed performance of the Topic Method (with 10 and 50 topics) and Tag Method for classifying individual tags, confirming our intuition about why the Tag Method is performing better than the Topic Method under the omission-penalizing metrics. Results shows that the Tag Method omits much fewer tags than the Topic Method, therefore, gaining precision scores for rarer tags (such as "meditation", "male_and_female" etc), for which the Topic Method receives zeros. The plots also show that the Topic Method and Tag Method are very similar in their precision, recall and F-1 performance for more common tags (e.g., "opera", "drums", "strings" etc), and that the model with more topics (i.e. 50) generally outperforms that with fewer topics (i.e. 10) on the same tags.

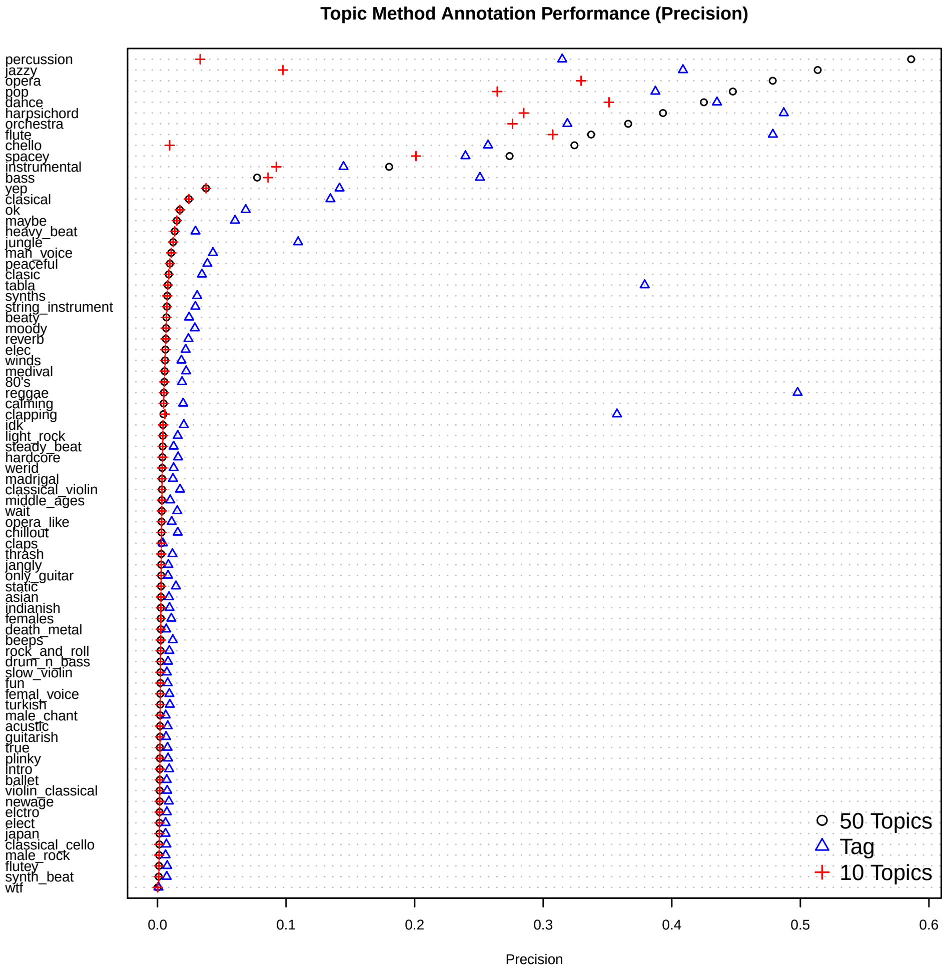


Figure 3.28: Precision

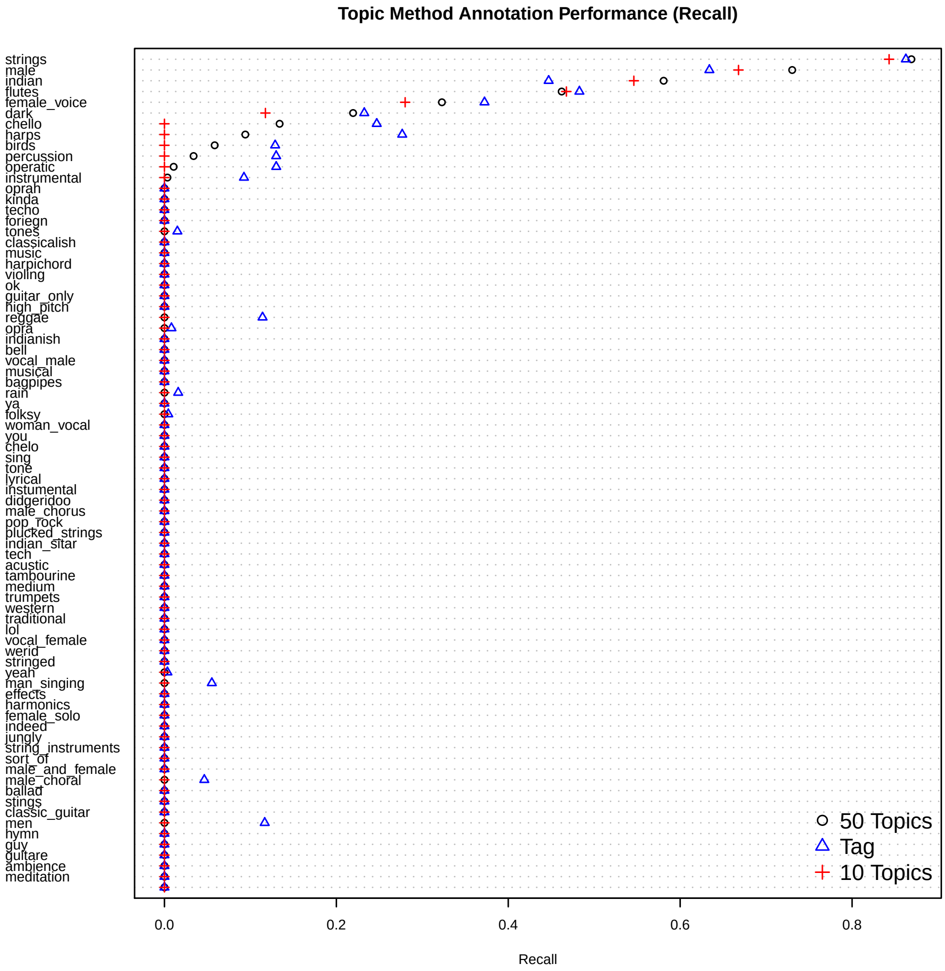


Figure 3.29: Recall

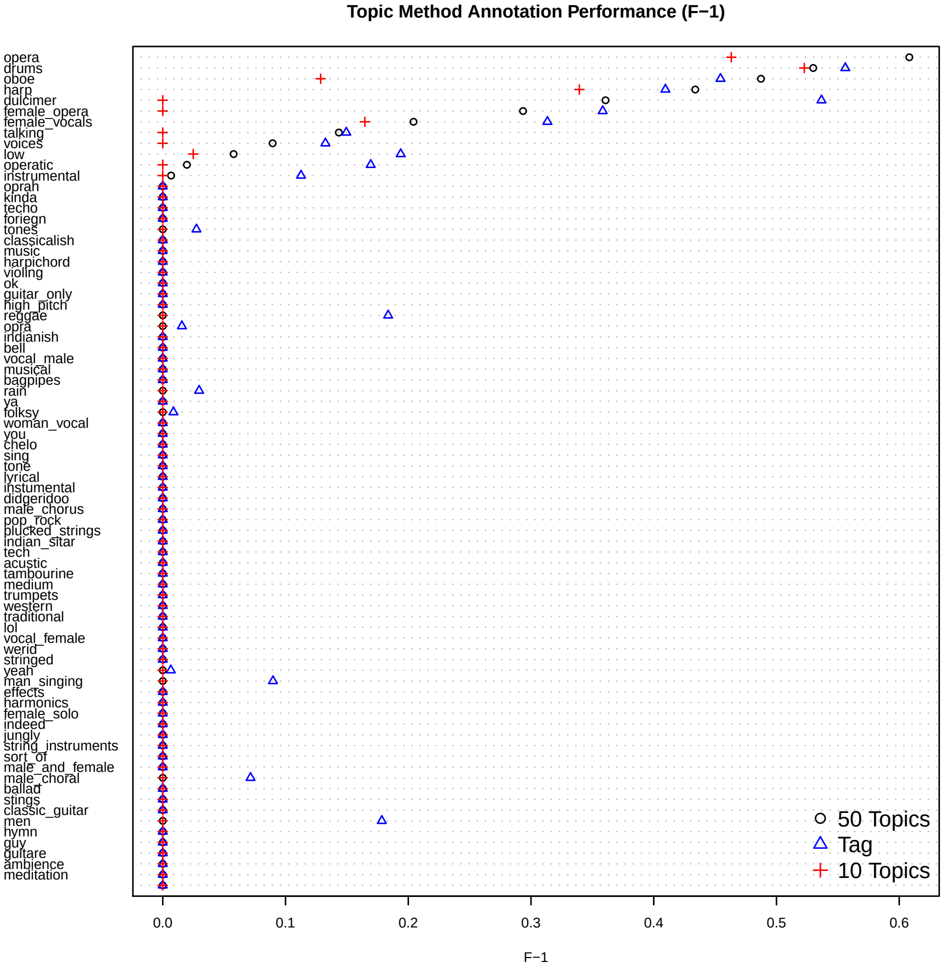


Figure 3.30: F-1

3.3.4.4 Experiment 3: Retrieval Performance

The tags generated by a music tagger can be used to facilitate retrieval. Given a search query, music clips can be ranked order by the KL divergence between the query tag distribution and the tag probability distribution for each clip. We measure retrieval performance using the mean average precision (MAP) [WikipediaMAPDefinition 2012] metric, which computes precision (the number of retrieved music clips whose ground truth tags include the search query) while placing more weight on the higher ranked clips.

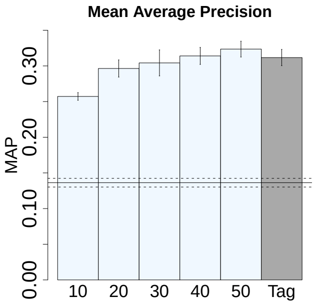


Figure 3.31: Retrieval Performance, in terms of average mean precision

Figure 3.31 shows the retrieval performance of the three methods under this metric. The retrieval performance of the Topic Method (with 50 topics) is indistinguishable from the Tag method, and both methods significantly outperform the random baseline.

3.3.4.5 Experiment 4: Efficiency

One of the main motivation behind using the Topic Method to generate tags is efficiency, i.e., it is much faster to train a classifier to predict 50 topic classes than 834 tag classes.

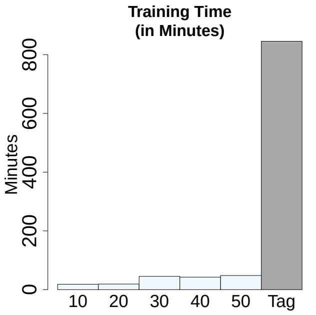


Figure 3.32: Comparison of efficiency in terms of training time

Figure 3.32 shows a rough estimate of the training time (averaged over folds) of the different models. While the training time does increase as the number of topics

(a) Missing Ground Truth Annotation

(i) sitar eastern
TP: indian sitar eastern guitar oriental strings middle_eastern slow drums arabic
TG: indian sitar guitar eastern slow drums oriental india strings solo
(ii) rock
TP: rock guitar male male_vocal pop loud man vocal metal drums
TG: rock male male_vocals male_vocal guitar male_voice pop loud man vocals
(iii) singing
TP: choir choral opera vocal vocals chorus chant singing female classical
TG: choir choral vocal opera singing chorus vocals female voices woman

(b) Vocabulary Mismatch

(i) faster jazzy beat fast disco guitar dance pop cymbals drums rock 80s upbeat electro
TP: electronic drums synth rock beat fast guitar dance electro beats
TG: electronic beat drums synth electro fast electric beats guitar rock
(ii) woman popish female_voice pop female vocal female_singer synth
TP: female woman vocal female_vocal female_voice singing pop vocals female_vocals voice
TG: female woman pop female_vocal singing vocals female_voice vocal guitar female_vocals
(iii) celtic classic violins violin medival strings
TP: classical violin guitar strings slow irish harp classic violins country
TG: classical strings violin classic guitar fiddle violins string baroque medieval

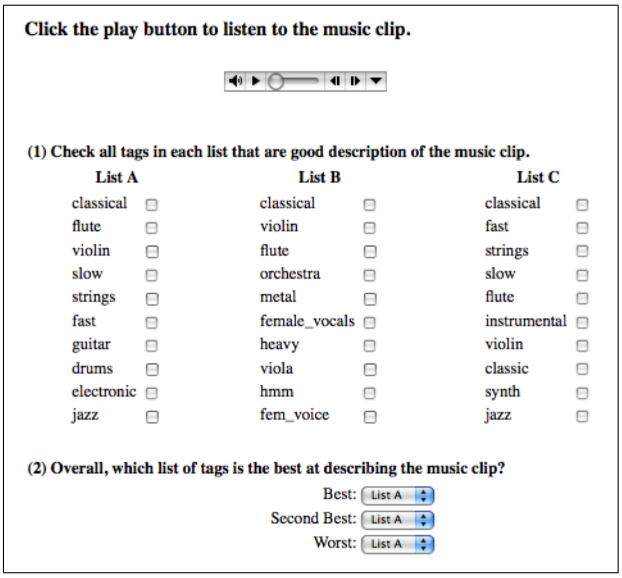
Table 3.10: In **bold** are the ground truth tags. TP and TG refers to the Topic Method and Tag Method respectively.

increases, the training time plateaus and are very similar for topic models with 30, 40 or 50 topics. The most important observation is that the Topic Method is approximately 94% times faster to train than the Tag Method, which confirms our belief that our proposed method will be significantly more scalable as the size of the tag vocabulary grows.

Experiment 5: Evaluation

The performance metrics we have used so far can only *approximate* the quality of the generated tags. The reason is that the ground truth tags collected by TagATune, which we are using as if they were gold standards, can never be fully complete. When deciding if a generated tag is accurate, comparison against the ground truth set will systematically under-estimate performance, due to missing tags or vocabulary mismatch.

Consider the examples in Table 3.10. Table 3.10(a) shows examples of music clips that have only one or two ground truth tags (in **bold**). In this case, generated tags that cannot be found amongst ground truth tags are counted as wrong, when in fact they are correct. For example, the tags “india”, “oriental”, “middle eastern” (example i), or “guitar”, “loud”, “drums” (example ii), or “vocal”, “chorus”, “chant” (example iii) are not considered correct tags, even though they are either equivalent in meaning to the ground truth tags, or highly correlated and likely to be correct (e.g., in the case of “drums” and “rock”). Figure 3.10(b) show examples where the ground truth set tags *do* provide sufficient coverage, but because of vocabulary



Click the play button to listen to the music clip.

(1) Check all tags in each list that are good description of the music clip.

List A		**List B**		**List C**	
classical	□	classical	□	classical	□
flute	□	violin	□	fast	□
violin	□	flute	□	strings	□
slow	□	orchestra	□	slow	□
strings	□	metal	□	flute	□
fast	□	female_vocals	□	instrumental	□
guitar	□	heavy	□	violin	□
drums	□	viola	□	classic	□
electronic	□	hmm	□	synth	□
jazz	□	fem_voice	□	jazz	□

(2) Overall, which list of tags is the best at describing the music clip?

Best: List A

Second Best: List A

Worst: List A

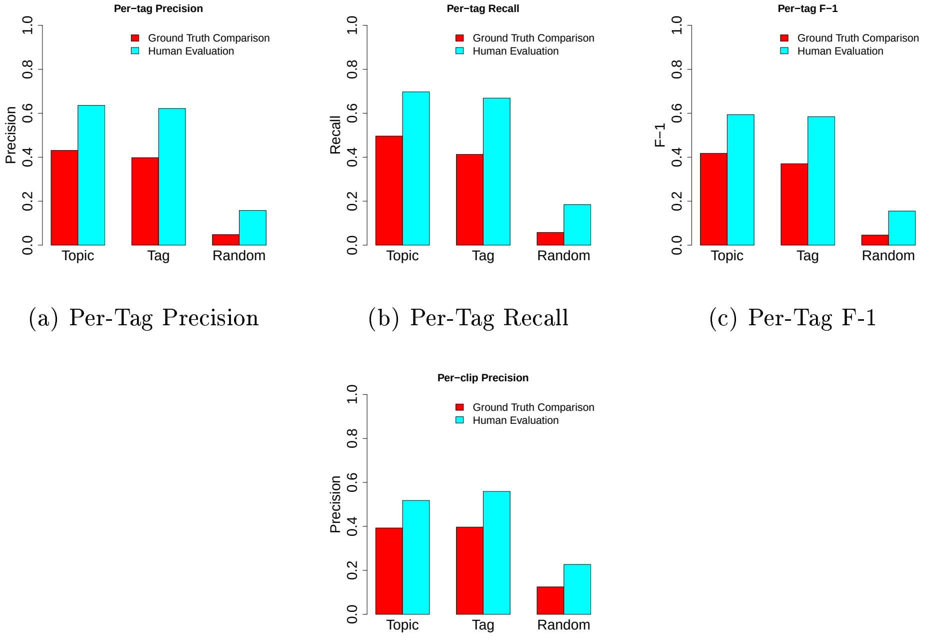
Figure 3.33: Mechanical Turk Annotation Experiment: Interface

mismatch, there are again many false negatives. Examples of vocabulary mismatch include “electro” versus “electronic”, “beats” versus “beat” (example i), or “female voice” versus “female vocals”, “pop” versus “popish” (example ii), or “celtic” versus “irish”, “medival” versus “medieval”, or “strings” versus “string” (example iii).

Annotation Performance

In order to compare the true merit of the competing approaches, we conducted a Mechanical Turk experiment where we ask humans to evaluate the tags generated by the Topic Method (with 50 topics), Tag Method and Random Method. We randomly selected a set of 100 music clips (20 in each fold) and solicited evaluations from 10 unique turkers for each music clip. For each clip, the turker is given three lists of tags of the same size, generated by the Topic Method, Tag Method, and the Random Method respectively. The order of the three lists are randomized to eliminate any presentation bias. The turkers are asked to (1) click the checkbox beside a tag if that tag is appropriate for the music clip (i.e. describes the music well), and (2) rank order the three lists based on how well they describe the music clip overall. Figure 3.33 shows the interface for the Mechanical Turk annotation experiment.

Figure 3.34 shows the per-tag precision, recall and F-1 scores as well as the per-clip precision scores of the three methods, when we evaluate tags by comparing to the ground truth set versus using human evaluation. Results show that when tags are judged based on whether they are found amongst the ground truth tags, the performance of the tagger is grossly underestimated under any metrics. In fact, of the tags (generated by either Topic Method or Tag Method) that the turkers considered as “appropriate” for any music clip, on average, approximately 50% of



(a) Per-Tag Precision

(b) Per-Tag Recall

(c) Per-Tag F-1

(d) Per-Clip Precision

Figure 3.34: Mechanical Turk Annotation Experiment: Results

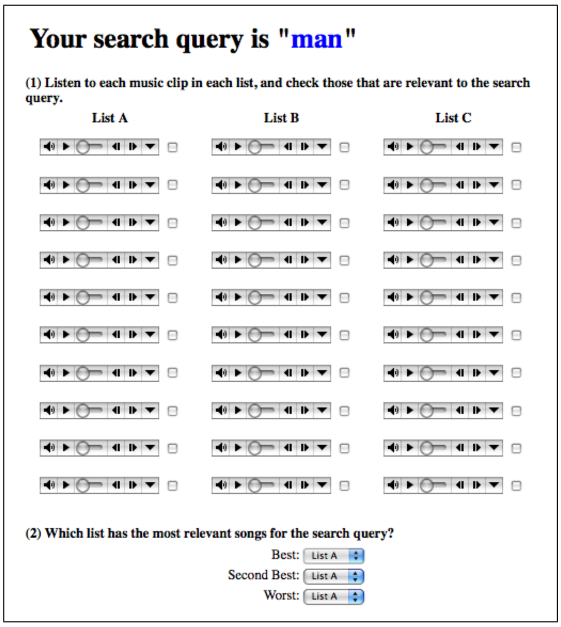
them are not found in the ground truth sets.

While the performance of Topic Method and Tag Method are similar in this experiment, when asked which *list* of tags the human user prefers the most, second most and the least, the average numbers of votes (out of 10) are 6.20 for the Tag Method, 3.34 for the Topic Method, and 0.46 for the Random Method, in strong favor of the Tag Method. Our hypothesis is that people actually prefer the Tag Method because its tag coverage is better than the Topic Method. This observation has interesting implications for how tags should be evaluated, individually versus as a whole.

Retrieval Performance

We conducted a similar experiment for evaluating retrieval performance. Figure 3.35 shows the interface for the Mechanical Turk annotation experiment.

Similar to the annotation task, our hypothesis is that the retrieval performance of the three methods, under the mean average precision metric, is underestimated because many of the retrieved music clips are false negatives if the search query cannot be found amongst their ground truth tags. To test this hypothesis, we ran a similar Mechanical Turk experiment, where we provide each turker a search query and three lists of music clips retrieved for that search query by the Tag Method, Topic Method and Random Method. Again, the order of the lists is randomized to



Your search query is "man"

(1) Listen to each music clip in each list, and check those that are relevant to the search query.

List A List B List C

(2) Which list has the most relevant songs for the search query?

Best: List A

Second Best: List A

Worst: List A

Figure 3.35: Mechanical Turk Retrieval Experiment: Interface

prevent presentation bias. There are 100 one-word queries in total, and 3 users for evaluating the music clips retrieved for each query. Users are asked to check the checkbox of each music clip that they consider "relevant" for the query. In addition, they are asked to rank order the three lists in terms of their overall relevance to the query.

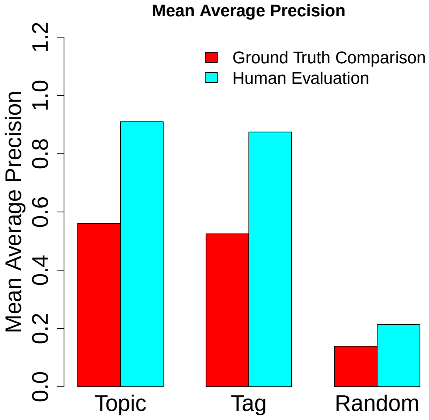


Figure 3.36: Mechanical Turk Retrieval Experiment: Results

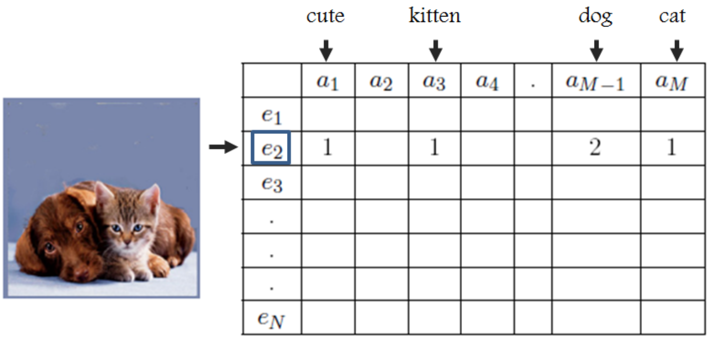
Figure 3.36 shows the mean average precision, when the ground truth tags versus human judgment is used to evaluate the relevance of each music clip in the retrieved set. Results show that when humans evaluate the retrieved list, the mean average precision of all methods are significantly higher than if we use the ground truth tags as the judge. Finally, when asked which list of music clips the turker prefers the most, second most and the least, the average numbers of votes (out of 3) are 1.17 for the Tag Method, 1.78 for the Topic Method, and 0.56 for Random Method, in favor for the Topic Method.

3.4 Conclusion

3.4.1 Reference to the Framework

Recall from our framework description in chapter 1, that the attribute learning problem can be described as the task of filling in a $N \times M$ matrix $\mathcal{V}$, where the rows are a set of entities e_i, $i = 1 \ldots N$, the columns are a set of attribute a_j, $j = 1 \ldots M$, and each cell of the matrix $v_{ij} = f_j(e_i)$ is the value of the attribute a_j for entity e_i.

In this chapter, we introduced a new game mechanism (and an implemented game called TagATune) for eliciting attributes for music. In general, one can think of *games with a purpose* as a **query model** for populating the matrix $\mathcal{V}$. For example, the query model of the ESP Game (where players enter tags until one of their tags match) chooses a single entity e_i (i.e., a row in the matrix) to present to two game players, and receives a set of (attribute, attribute value) tuples $\{(a_j, v_{i,j})\}$, where $v_{i,j}$ is the number of times the attribute a_j has been associated with entity e_i by different players.



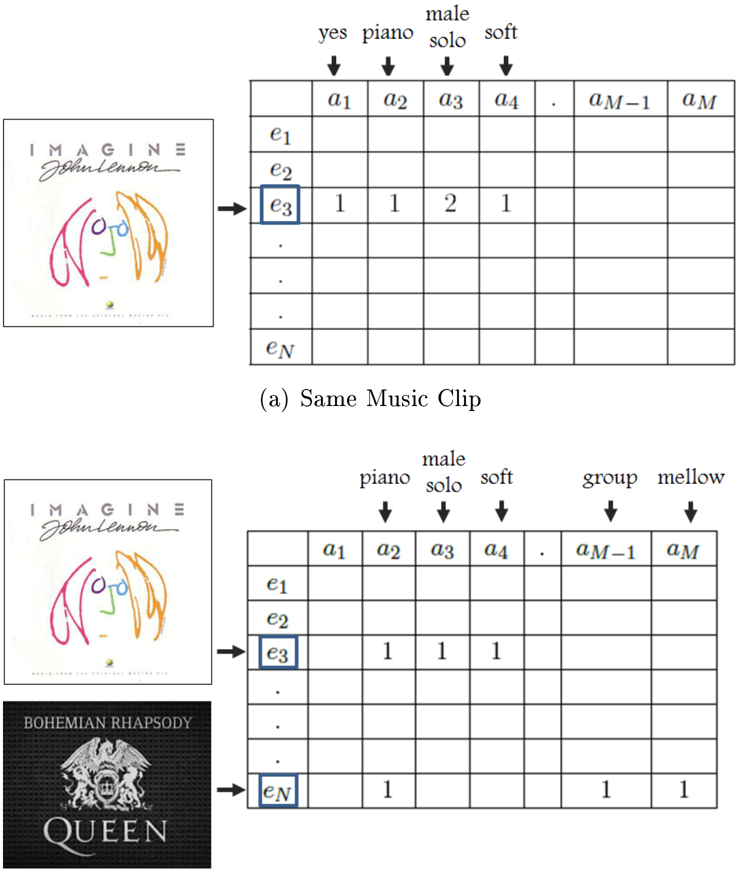
	a_1	a_2	a_3	a_4	.	a_{M-1}	a_M
e_1							
e_2	1		1			2	1
e_3							
.							
.							
.							
e_N							

Figure 3.37: Query Model of the ESP Game

.

Figure 3.37 shows an example of the query model of The ESP Game. Here, the query selects image e_2 and presents it to two players. One of the players entered the tag “cute,” “kitten” and “dog”, while the other player entered the tag “cat” and “dog”. Based on the tag counts, the query receives {(cute, 1),(kitten 1),(dog 2),(cat 1)} from the two players.

In the TagATune game, the query model is different. In the case where the music clips are the same, the system selects one music clip e_i to present to both players, and receives a set of (attribute, attribute value) tuples $\{(a_j, v_{i,j})\}$, where



(a) Same Music Clip

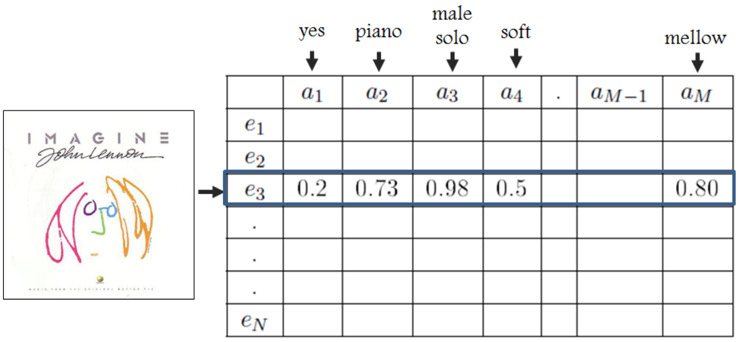
(b) Different Music Clips

Figure 3.38: Query Model of TagATune

$v_{i,j}$ is the number of times the attribute a_j has been associated with the music clip e_i by different players. For example, Figure 3.38(a) shows an example where the two players were given the same music clip (John Lennon's *Imagine*). One player enters "male solo," "soft," "yes" (in response to the other player's tag "piano"), while the other player enters enters "male solo" and "piano." In the case where the music clips are different, the system selects two different music clips e_i and e_k, one for each of the two players, and receives two sets of (attribute, attribute value) tuples $\{(a_j, v_{i,j})\}$ and $\{(a_j, v_{k,j})\}$. Figure 3.38(b) shows an example where the two players were given different music clips (John Lennon's *Imagine* and Queen's *Bohemian Rhapsody*). The player given a music clip from *Imagine* entered the tags "piano," "male solo," "soft", while the other player entered the tags "piano," "mellow," "group."

From these two examples, one can see why TagATune is an appropriate game mechanism for extracting attributes in a vocabulary-rich setting, where the input data (in this case, music) can be associated with a huge number of columns in the matrix. Here, it is much more difficult for players to find the same column of the matrix if the game is implemented using the output-agreement (the ESP Game) mechanism. Furthermore, by allowing communication and rewarding players for successfully determining whether their music clips are the same or different, players are motivated to enter tags that *discriminate* between two pieces of music, instead of ones that their partner is likely to type. As the result, the tags that we retrieve from TagATune are much more informative and specific.

From the same two examples, however, we observe that the data collected from TagATune data presents some challenges if it were to be used for training machine learning algorithms to predict tags. For example, some attributes, e.g., “yes,” are irrelevant to the content of the music, while others, e.g., “mellow” and “soft,” are redundant, causing the training examples for “mellow,” “soft” and other similar attributes to be split amongst these synonymous classes. Traditionally, machine learning algorithms are trained on a small number of mutually exclusive labels: the learner is given a small matrix where the columns are partially filled in, and the goal is to predict the rest of the values of the matrix. In contrast, here the columns can be redundant, impossible to learn (e.g., in the case of the attribute “yes”), and the matrix itself can be huge in terms of the number of columns. In this chapter, we presented a technique that makes use of this noisy data collected by TagATune to train a music tagger efficiently.



	yes	piano	male solo	soft			mellow
	a_1	a_2	a_3	a_4	.	a_{M-1}	a_M
e_1							
e_2							
e_3	0.2	0.73	0.98	0.5			0.80
.							
.							
.							
e_N							

Figure 3.39: Using TagATune for Evaluation

.

The data collected from TagATune also creates a very sparse matrix, with many missing attributes values. Using the TagATune data as ground truth for evaluation poses a problem: just because an attribute value is missing does not mean that that attribute does not apply to the music clip. For example, if a music tagging algorithm predicted *Imagine* to be “mellow,” it should be marked as correct even though no previous users have associated that particular tag to the music clip. The idea of using TagATune for evaluation is illustrated in Figure 3.39: the game serves music attributes generated by an algorithm (in the order of their probability) and the strength of the algorithm is measured by the percentage of players who are able to guess correctly that the music clips given to them and the algorithm are the same.

3.4.2 Lessons Learned

Until recently, efforts of research in human computation have largely been centered around the development of games that follow existing mechanisms, with less focus on the invention of new game mechanisms for data collection. The main contribution of this chapter is the introduction of the input-agreement mechanism, a new method for collecting data in human computation games. We developed a game called TagATune that uses this new mechanism to collect tags for music and sound clips, and presented statistics on the data collected during the seven-month period after the game was launched. Results show that the popularity and throughput of TagATune are superior to other human computation games for collecting music metadata. Moreover, this new mechanism is readily extensible to images and videos.

In re-designing TagATune to be more human friendly, however, we introduced problems: the data collected by TagATune is extremely noisy and unamenable to training standard machine learning algorithms for music tagging. We created a machine learning algorithm to make use of the vast, open vocabulary data extracted from TagATune. In particular, we showed that topic models can be used to define a reduced set of labels, using which the task of mapping from audio features to tags is made more efficient. We compared the Topic Method and Tag Method on five criteria: feasibility, annotation performance, retrieval performance, computational efficiency, and annotation and retrieval performance as judged by human evaluators. Our main results show that our proposed method is feasible, both data-efficient (i.e., can utilize an arbitrary open vocabulary of tags) and time-efficient (i.e., reduces training time by 94% compared to learning from tag labels directly), and achieves comparable performance for annotation and superior performance for retrieval.

This work reveals the delicate balance between designing the system to achieve the *human-centric* versus *task-centric* objectives, and illustrates our approach to human computation: to first design the system to achieve the *human-centric* objective, then tackle the task-centric objective with the help of machine learning algorithms.

Chapter 4

Learning Attributes under Knowledge Limitations

4.1 Overview

Many attribute learning tasks distributed on crowdsourcing platforms today are *perceptual tasks* that require only common sense knowledge and no special expertise. In other words, these tasks can be tackled equally well by anyone who can see and hear – most people are equally competent at identifying "bicycles" and "cows" in an image, or describing a piece of music as "soothing," "slow" and containing "female vocals". In fact, most, if not all, human computation games are built to distribute perceptual tasks that are difficult for machines, but trivial for any average game player from the general population.

Not all attributes, however, are easy to determine. In particular, identifying the category of an entity – e.g., whether the bird in an image is a *Pied-Billed Grebe*, whether a piece of music belongs to the genre *blues* or *jazz*, whether Jon Kyl is a *politician*, whether a piece of art is *impressionist* or *expressionist* – usually requires special knowledge about either the entity itself (e.g., knowing who Jon Kyl is) or how attributes map to categories (e.g., knowing that Grebes are usually "found on water", have "lobed toes", are "plain-colored in dark browns and whites.") In analyzing the TagATune data, we encountered some evidence of this phenomenon; for example, there exist game partners where one player entered the tag "harpsichord", while the other player responded with the question "what is a harpsichord?" Knowledge limitations may not be as problematic in paid crowdsourcing setting, where workers can be given additional time and resources (e.g., a search engine) to look up information to perform a knowledge-intensive task. In contrast, knowledge limitations can be detrimental in a game setting, where players' perception of how enjoyable the game is depends highly on the degree to which they can successfully accomplish the tasks posed by the game. Moreover, causal games are usually fast-paced; expecting players to read up extensive amount of information in order to play the game is simply infeasible.

In this chapter, we explore how to design a hybrid human and machine computation system for learning attributes under knowledge limitations, e.g., when workers are unfamiliar with the attributes or the entities themselves. To explore this problem, we will first present a new game mechanism called *complementary-agreement* for extracting attribute and attribute values, and evaluate its effectiveness using two case studies – the first case study involves bird image categorization in the casual game setting, while the second case study involves noun phrase categorization in the paid crowdsourcing setting.

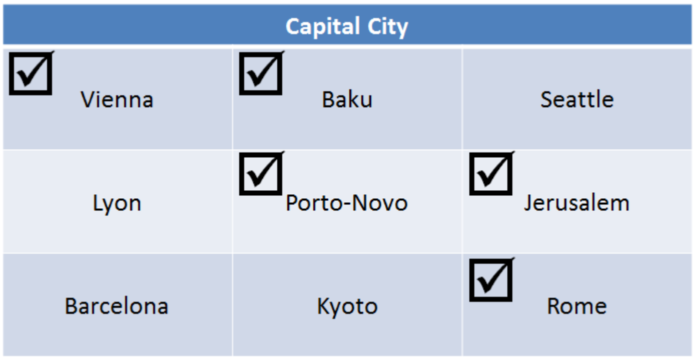
4.2 Complementary-Agreement Mechanism

Human computation games like the ESP Game and TagATune ask open-form question, where players are allowed to describe the entity (i.e., images or music) freely, without having to answer in a way that adheres to any vocabulary or language restrictions. Categorization, on the other hand, is usually concerned with classifying entities into a set of pre-defined categories; hence, closed-form questions, which present an entity along with a fixed set of categories, are more appropriate.

Categorization tasks can take many formats. Suppose that there are N entities and M categories, the most common formats are:

- *single-entity, binary-choice* query, which presents one entity e_n and one category c_m and ask if the entity belongs to that category.
- *single-entity, multiple-choice* query, which presents one entity e_n and multiple categories c_1, c_2, $\ldots$ c_K, $K \leq M$, and ask if the entity belongs to one (or several) of the categories.

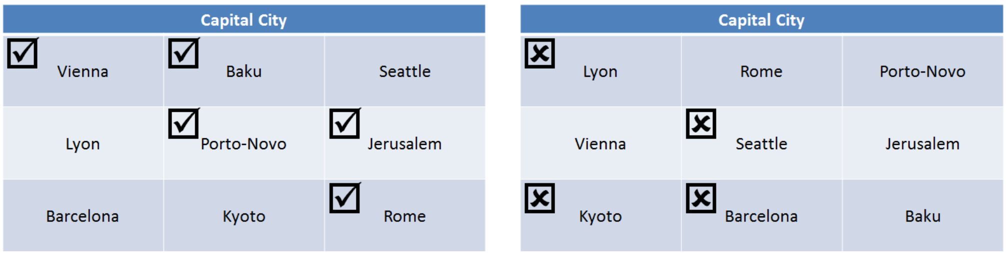
An alternative and less common format is the *multiple-entity, binary-choice* query, which presents one category c_m and multiple entities e_i, $\ldots$ e_K, $K \leq N$, and ask which entity belongs to that category. An intuitive interface for asking this type of query is show in Figure 4.1, where the worker is presented with a category and a grid of entities and asked to select the entities in the grid that belong to that category.



Capital City		
☑ Vienna	☑ Baku	Seattle
Lyon	☑ Porto-Novo	☑ Jerusalem
Barcelona	Kyoto	☑ Rome

Figure 4.1: Multiple-Entity, Binary-Choice Query in a Grid

The *multiple-entity, binary choice* query has the advantage of asking many binary questions in a single query. However, it raises new questions about how to verify workers' answers for this type of query. Specifically, when presented on a paid crowdsourcing platform, how can we prevent *lazy* workers from selecting as few entities as possible, or *spam* workers from selecting entities at random or all the entities all the time? When presented in a human computation game, what mechanism is appropriate for ensuring that players are motivated to select only those entities that belong to the category, no more and no less?



Capital City		
☑ Vienna	☑ Baku	Seattle
Lyon	☑ Porto-Novo	☑ Jerusalem
Barcelona	Kyoto	☑ Rome

(a) Positive Worker

Capital City		
☒ Lyon	Rome	Porto-Novo
Vienna	☒ Seattle	Jerusalem
☒ Kyoto	☒ Barcelona	Baku

(b) Negative Worker

Figure 4.2: Complementary-Agreement Mechanism

The method we propose here is called the *complementary-agreement* mechanism (Figure 4.2). In this mechanism, a pair of workers are shown a categorical attribute (e.g., is a *capital city*) and a grid of entities (e.g., names of cities) in randomized order, and asked to split the entities into two sets – one set of entities that belong to that category, and the other set of entities that do not belong to that category. To encourage workers to select *only* the entities they think belong to the category, the mechanism gives the query to two different workers, and asks one of the workers (which we refer to as the *positive* worker) to select entities that can be described by that categorical attribute, and the other worker (which we refer to as the *negative* worker) to select entities that cannot be described by that categorical attribute. Each workers is required to select at least one entity from the grid, and is rewarded for selecting as many entities as possible while *not* overlapping with the other worker's selections.

The complementary-agreement mechanism elicits truthful outputs by explicitly preventing players from entering outputs that the system considers undesirable. This game mechanism was inspired by the game KissKissBan [Ho 2009], which adds a third player (called the *blocker*) to the ESP Game, who acts as an adversary that enters tags to block the other two players from matching. Like the KissKissBan game, in the complementary-agreement mechanism, players are generating *complementary* data that is used to constrain each other's outputs. As a result, the positive player is motivated to select only entities that truly belong to the category, in order to minimize the chances of overlap with the negative player's outputs, vice versa.

Applied to a categorization task, the complementary-agreement mechanism essentially asks workers to generate the values of a categorical attribute for the given set of entities, where the value is 1 if an entity belongs to that category, 0 otherwise. Beyond eliciting the values of categorical or non-categorical attributes, the complement-agreement mechanism can be used to *discover new attributes* from workers, e.g., by asking one of the workers to enter an attribute a that split the set of entities, then asking both workers to determine the value of a for each entity in the grid. In the next section, we will present a human computation game that makes use of these two different modes of the complementary-agreement mechanism.

4.3 Case Study I: Bird Image Classification

4.3.1 Motivation

In the first case study, we are interested in using the complementary-agreement mechanism for classifying bird images within the context of a human computation game. Bird classification is a task well suited for our inquiry – visually identify the correct species of a bird requires specialized knowledge, and doing so in a human computation games, where the average player may not be an avid birder, is especially challenging. This begs the questions of whether there exist a hybrid approach, involving both humans and machines, that can help overcome humans' knowledge limitation in bird image categorization.

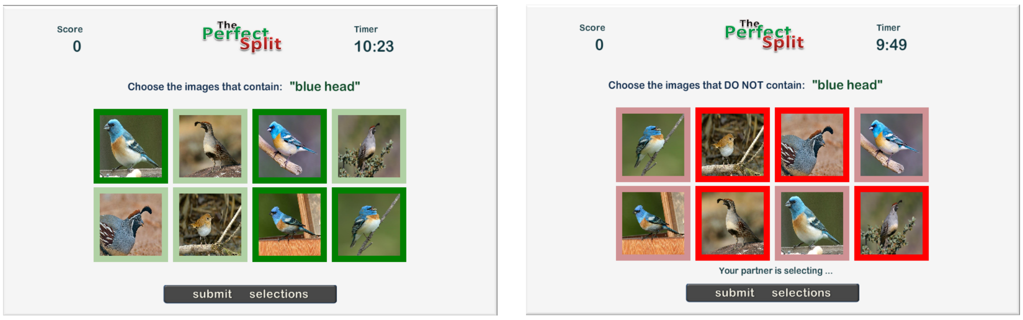
In this section, we will describe the design of a new game called *The Perfect Split*, which implements the complementary-agreement mechanism, and how it can be used to extract attributes and attribute values for bird images. Through a user study involving 30 game players, we compare two approaches for classifying bird images by species – the *human categorization* approach, where players directly classify bird images by category, and the *hybrid categorization* approach, where the game asks players to provide perceptual attributes and attribute values, then uses the collected data as features to train a machine learning algorithm to perform the actual categorization.

4.3.2 The Perfect Split

The Perfect Split is a two-player image annotation game. In this game, two players are shown a grid of images, and asked to split them into two sets – one set of bird images that has a particular attribute, and the other set of bird images that do not have that attribute. Players alternate between being the *positive* player, who selects images that have a particular attribute, or the *negative* player, who selects the images that do not have a particular attribute. Attributes can be categorical (e.g., "Owl," "Western Tanager") or non-categorical (e.g., "blue head," "white spot on wing"), pre-specified by the game or entered by game players.

4.3.2.1 Game Modes

The game consists of two distinct modes: attribute scoring mode and attribute discovery mode. In the **attribute scoring mode** (Figures 4.3), the game provides the players with an attribute. Players simply have to select the images that have (or do not have) the given attribute. Players can select or unselect the images by clicking on them, and submit their selections when they are ready.



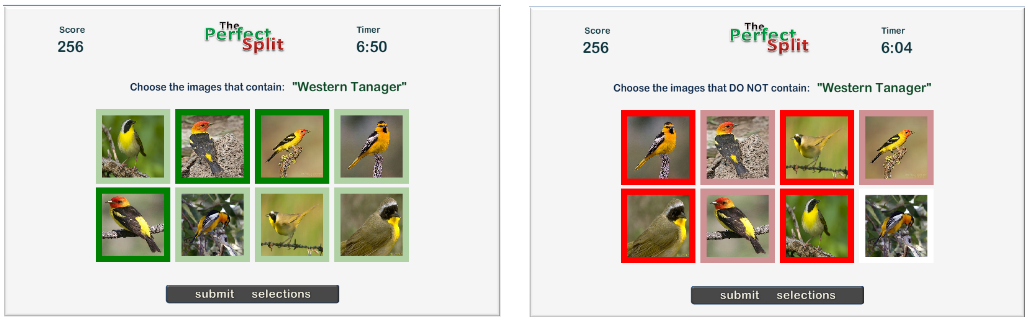
Choose the images that contain: "blue head"

Choose the images that DO NOT contain: "blue head"

Your partner is selecting ...

(a) Positive Player (b) Negative Player

Figure 4.3: The Perfect Split: attribute scoring mode for a non-categorical attribute



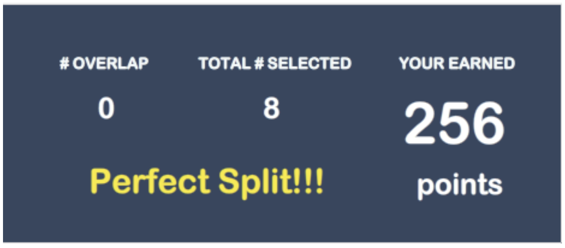
Choose the images that contain: "Western Tanager"

Choose the images that DO NOT contain: "Western Tanager"

(a) Positive Player (b) Negative Player

Figure 4.4: The Perfect Split: attribute scoring mode for a categorical attribute

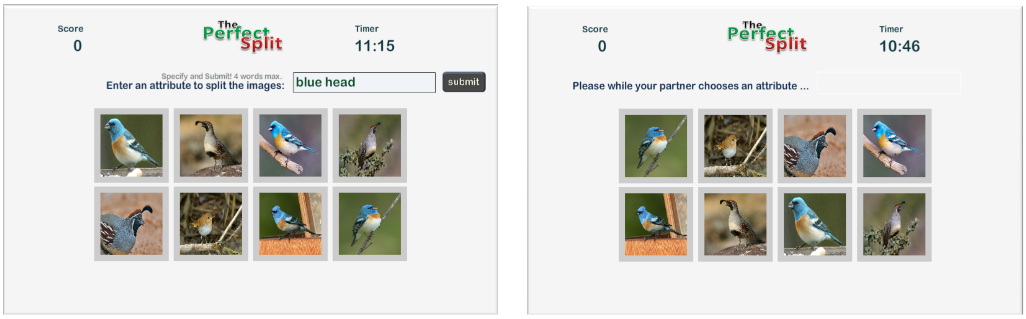
At the end of each round, the game provides feedback to both players (Figure 4.5), including (i) the total number of images selected by the two players, (ii) the number of overlap selections, (iii) the round score, and (iv) a message to indicate to players how well or poorly they performed in that round (e.g., “Terrible,” “Just OK,” “Fantastic,” “You Rock,” “Perfect Split”).



# OVERLAP	TOTAL # SELECTED	YOUR EARNED
0	8	256
Perfect Split!!!		points

Figure 4.5: End of Round Feedback

In the **attribute discovery mode**, the attribute is not provided by the game; instead, one of the players (specifically, the *positive* player) enters an attribute that split the images. The player-generated attribute is then revealed to both players, who then proceed to the scoring mode to select images that do (or do not) have



Specify and Submit! 4 words max.
Enter an attribute to split the images: blue head

Please while your partner chooses an attribute ...

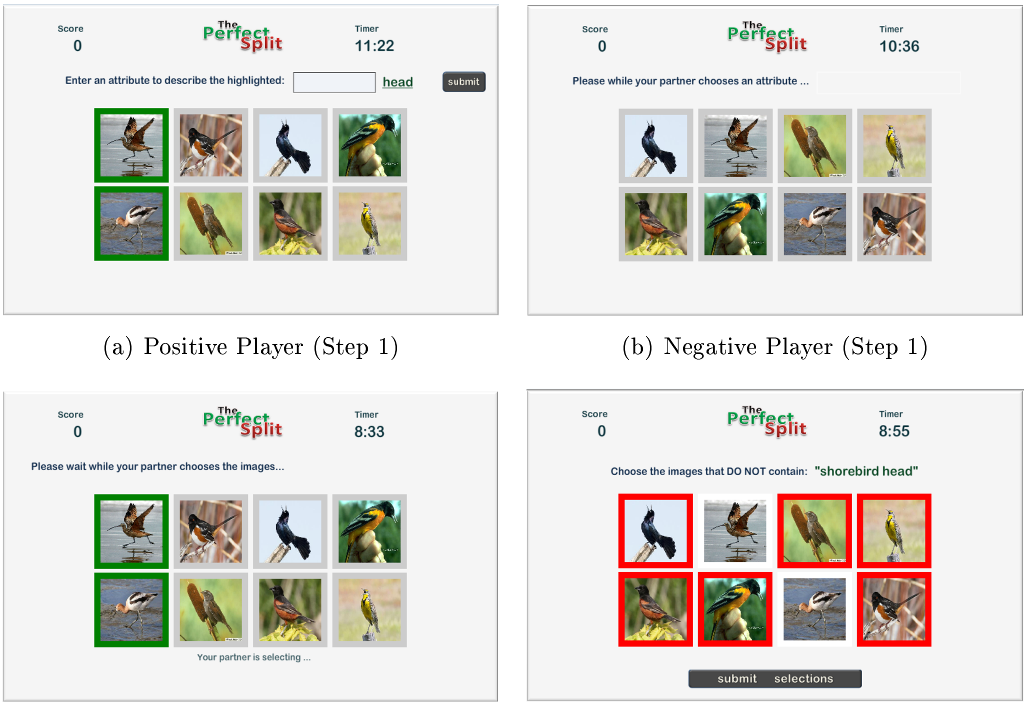
(a) Positive Player

(b) Negative Player

Figure 4.6: The Perfect Split: Discovery Mode

that particular attribute. Figure 4.6 shows the interface for the discovery mode of the game.

There is a slight variation of the discovery mode, called the **targeted discovery mode**, where the positive player is shown a set of highlighted images, and is asked to enter an attribute to distinguish the highlighted images from the other images. Upon seeing the positive player's attribute, the negative player selects the images that do not have that attribute. Figure 4.7 shows the two steps process of the targeted discovery mode.



Enter an attribute to describe the highlighted:

Please while your partner chooses an attribute ...

(a) Positive Player (Step 1)

(b) Negative Player (Step 1)

Please wait while your partner chooses the images...

Your partner is selecting ...

Choose the images that DO NOT contain: "shorebird head"

(c) Positive Player (Step 2)

(d) Negative Player (Step 2)

Figure 4.7: The Perfect Split: Targeted Discovery Mode

4.3.2.2 Vocabulary Restrictions

In the discovery mode, the game prompts the positive player to enter an attribute in two different ways. In some rounds, the player is provided with a textbox where he or she can enter any attribute up to 25 characters in length (e.g., Figure 4.6(a)). In other rounds, the player is given some restrictions on the types of attributes they can enter (e.g., Figure 4.7(a)); specifically, they are provided with a textbox that imposes a stricter character limit (i.e., 15 characters), followed by the name of a commonly known body part, such as head, wing, breast, belly, back and beak. In this case, the positive player is asked to enter an attribute to describe that particular body part of the bird.

actual or similar bird family and species
owl, bald eagle, waterbirds, pigeon-like, ducks
eyes
red circle around eye, black eye
chest
birds with spotted upper chest
visibility of body parts
bird can see its back, birds can see their back
wings
brown dotted wings
belly
white belly, bird with spotted belly
legs
long legs, red feet
feathers
brownish grey feathers, yellow feathers, red feathers, grey and white feather, black feathers, blue feather, red feather, brownish feathers, blue feathers, birds with yellowish feathers, birds with black feathers,
head
black and white head, top of head brown and white, black head, white head, red head, pure white heads, no black head, birds with a red head, blue head
beak
thin long beaks, long beak, long thin beak, yellow beak, red mouth, blue beaks
general colors
yellow, black bird, blue bird, birds with brown shade, birds having yellow, yellow birds, no yellow, brown, is completely black, birds with yellow, no yellow in image, not black, have YELLOW color, no blue color, almost white all over, bluish birds, all yellow birds, blue birds, black and white, red birds, blue colored bird, almost entirely bright blue or black, brown, monochrome, yellow body
poses, actions and locations
facing camera, fully front facing, beak open, back of bodies facing camera, can swim, birds walking, standing on a twig or limb, on tree limb, not STANDING, swimming, not in the water, not flying, standing on a tree, flying, standing in water, sitting on the ground, in water, ponds, birds in flight, birds on water, birds swimming, bird sitting on the floor, bird flying, swimming birds, birds walking on the ground, grass and sitting on a tree, birds in water
compound attributes
birds that don't have any red or pink in picture, hummingbirds/birds with red neck and long beak, grey bird on the ground and on cactus and on a tree, brown body black feathers, sparrows on trees, thin beak no red head, ducks in water, white stomach and dark color on the back, grey bird on the ground, ogange feather and facing left, long tail without feathers, brown head white body
other
cactus-like, white background, two birds

Table 4.1: Examples of Collected Attributes in Game Pilot

In an informal pilot study involving 10 users, we found when players were given no restrictions, the vocabulary of the collected attributes is extremely diverse. Table 4.1 illustrates this diversity: while players do describe specific body parts (where color is predominantly the attribute used in their description), more often, they describe the general colors, poses, actions and locations of the bird, or other attributes irrelevant to classification, such as "white background" and "two birds." Some body parts, such as feather, head and beak, are described more often than other body parts. The same attributes are often described in different ways, e.g., "blue bird," "blueish blue," and "blue colored birds."

In the current design of the game, we adopted the approach of imposing vocabulary and attribute length restrictions some of the times, while allowing players the freedom to enter arbitrary, lengthier descriptions at other times. Our goal is two-fold – to reduce the diversity of the vocabulary, and to encourage players to describe a variety of body parts, even those that are less salient (e.g., belly, wings, legs).

4.3.2.3 Selection of Images

There are two considerations when selecting the set of images to present to players in each round of the game. First, the system decides randomly whether to present a small number (3) versus a large number (5-8) of image clusters, where a cluster contains birds that all belong to the same category. Second, the system decides randomly whether to draw images based on coarse grain categories (e.g., family-level categories such as Owls versus Ducks) or fine-grained categories (e.g., species-level categories such as Snowy Owl versus Barn Owl). This variation in the granularity and in the number of clusters allows for varying levels of difficulty in the game, and prevents players from being able to guess the answer by anticipating the composition of the images in the grid.

4.3.3 User Study

We conducted a user study involving 30 participants playing The Perfect Split. Our primary objective is to understand the differences between two approaches to categorization – the *human categorization* approach, where players classify bird images directly, and the *hybrid categorization* approach, which asks players to generate and score perceptual attributes, then uses these perceptual attributes (and their values) as features to train a machine learning algorithm to perform the actual classification. In particular, we are interested in how these two approaches compare in terms of the extent to which they can achieve the *human-centric* object (i.e., the game is fun and easy for players) and *task-centric* objective (i.e., the game performs accurate categorization of bird images) of the system.

Family	Examples
Blackbird (15)	Baltimore Oriole, Boat-tailed Grackle, Bobolink, Brewers Blackbird
Bunting (3)	Indigo Bunting, Lazuli Bunting, Painted Bunting
Chickadee (4)	Black-capped Chickadee, Carolina Chickadee, Chestnut-backed Chickadee
Corvid (7)	American Crow, Blue Jay, Chihuahuan Raven, Common Raven, Fish Crow
Dove (7)	Band-tailed Pigeon, Common Ground Dove, Eurasian Collared Dove
Duck (20)	Red-breasted Merganser, American Wigeon, Black-bellied Whistling Duck
Eagle (3)	Bald Eagle, Golden Eagle, Osprey
Finch (12)	American Goldfinch, Black-headed Grosbeak, Common Redpoll
Flycatcher (12)	Ash-Throated Flycatcher, Black Phoebe, Eastern Kingbird, Eastern Phoebe
Grebe (3)	Eared Grebe, Pied-billed Grebe, Western Grebe
Gull (6)	California Gull, Great Black Backed Gull, Herring Gull, Laughing Gull
Hawk (10)	American Kestrel, Coopers Hawk, Merlin, Northern Harrier, Peregrine Falcon
Heron (11)	Black-Crowned Night Heron, Cattle Egret, Great Blue Heron, Great Egret
Hummingbird (5)	Allens Hummingbird, Annas Hummingbird, Black-chinned Hummingbird
Mimic (5)	Brown Thrasher, California Thrasher, Curve-Billed Thrasher, Gray Catbird
Nuthatch (3)	Pygmy Nuthatch, Red-breasted Nuthatch, White-breasted Nuthatch
Owl (5)	Barn Owl, Barred Owl, Eastern Screech Owl, Great Horned Owl, Snowy Owl
Quail (4)	California Quail, Gambels Quail, Northern Bobwhite, Scaled Quail
Shorebird (17)	American Avocet, American Woodcock, Black-Bellied Plover
Small-Dull (8)	Black-crested Titmouse, Blue-gray Gnatcatcher, Bushtit
Sparrow (20)	American Pipit, American Tree Sparrow, California Towhee, Canyon Towhee
Swallow (6)	Barn Swallow, Cliff Swallow, Northern Rough-winged Swallow
Tanager (3)	Scarlet Tanager, Summer Tanager, Western Tanager
Thrush (8)	American Robin, Eastern Bluebird, Hermit Thrush, Swainsons, Veery
Vireo (6)	Huttons Vireo, Red-eyed Vireo, Warbling Vireo, White-eyed Vireo
Warbler (17)	American Redstart, Black-and-white Warbler, Black-throated Green Warbler
Woodpecker (11)	Acorn Woodpecker, Downy Woodpecker, Golden-fronted Woodpecker
Wren (5)	Bewicks Wren, Cactus Wren, Carolina Wren, House Wren, Marsh Wren

Table 4.2: Composition of the Dataset

4.3.3.1 Dataset

The dataset consists of 1850 bird images belonging to 236 species and 28 families. One image per species is manually extracted from the allaboutbirds.org website; these images are curated by expert birders and therefore highly reliable. For each species, we extracted more images from the Web automatically, by querying for the species names using Google Image Search. Table 4.2 shows the composition of the dataset, including the number of species (in bracket) and a few examples of the species in each bird family.

4.3.3.2 Users

We recruited 30 participants through the Center for Behavioral and Decision Research (CBDR), a service provided by Carnegie Mellon University for recruiting subjects for experiments. Participants vary in ages (ranging from 20 to 62 years old), ethnic, socioeconomic and educational background. Each session involves two participants, who play The Perfect Split together as game partners. At the beginning of the session, participants were given a five-minute explanation of the game, and were told to not to verbally communicate with their game partner or use any external resources (e.g., search engine) to help them play the game. Each session lasts for roughly one hour, and each participants is paid $10 for their participation in the experiment.

4.3.3.3 Experimental Design

The session is divided into four parts. In each part, participants play a particular version of the game for 12 minutes, then fill in a short survey about their experience playing that version of the game. The four versions of the game (with their abbreviated names in brackets) include:

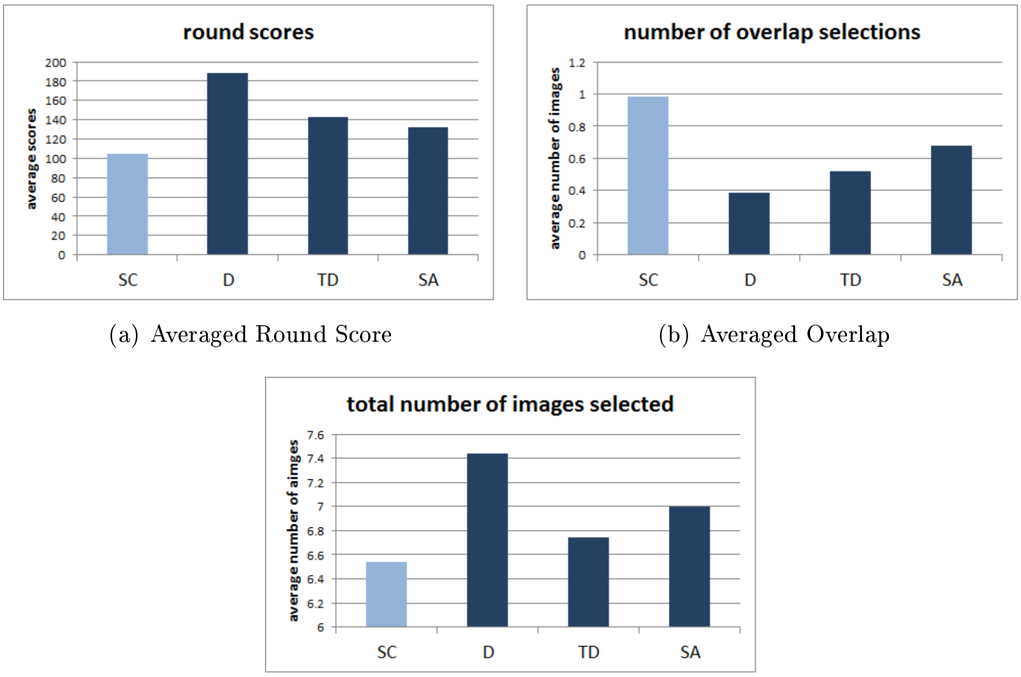
- **Scoring Mode / Categorical Attributes (SC)**: each round of the game is in the scoring mode; players are provided with a category name as the attribute and a set of bird images some of which are known to belong to the category.
- **Discovery Mode(D)**: each round of the game is in the discovery mode; the positive player is provided with no attribute, and asked to specify an attribute to split the images.
- **Targeted Discovery Mode (TD)**: each round of the game is in the targeted discovery mode; the positive player is provided with some highlighted images and no attribute, and asked to enter an attribute to distinguish the highlighted images from other images.
- **Scoring Mode / Player-Generated Attributes (SA)**: each round of the game is in the scoring mode; players are provided with an attribute that has been entered by players of previous games.

Our experiment follows a within-subject design, where each participant played all four versions of the game in one session. Within-subject design has the advantage of yielding more statistical significant results; however, there may be carryover effects if all subjects are asked to play the four versions of the game in the same order. To minimize these effects, we randomized the order of the game versions that participants play in each session, such that no two pairs of participants were exposed to the same ordering.

4.3.3.4 Results

The *human categorization* approach refers to the SC (Scoring Mode / Categorical Attributes) version of the game, where players were asked to select images that belong (or do not belong) to a category; in other words, this approach asks players to categorize bird images directly. The *hybrid categorization* makes use of a combination of the other three more open-ended versions of the game – namely, D (Discovery Mode), TD (Targeted Discovery Mode) and SA (Scoring Mode / Player Generated Attributes) – to collect perceptual attributes and attribute values from players, and uses a simple maximum entropy algorithm to predict the categories of the bird images using these player-generated attributes.

As mentioned before, our primary objective is to understand how these two approaches achieve the *human-centric* versus *task-centric* objectives of the system. In terms of the human-centric objective, we evaluate each version of the game in terms of how much players enjoyed their experience and how well they completed each round of the game. In terms of the task-centric objective, we compared the two approaches in terms of their classification performance across 28 different bird categories.



(a) Averaged Round Score

(b) Averaged Overlap

(c) Averaged Selection Total

Figure 4.8: Game Statistics Across Four Game Versions

Results (Figure 4.8(a), 4.8(b), 4.8(c)) show that players are much less successful in the game in the SC version, where they were asked to score categorical attributes directly, than the other three versions (D, TD and SA) of the game, where they were asked to generate perceptual attributes and attribute values. Specifically, in the SC version of the game, players have the lowest round scores ($p = 5 \times 10^{-5}$), greatest number of overlapped selections ($p = 1.8 \times 10^{-4}$) and the least number of images selected ($p = 4.6 \times 10^{-5}$), averaged first over the rounds of each game, then over all the games belonging to the each version. SC is colored in light blue, while the other three versions are colored in dark blue to show the distinction between the human (light blue) versus hybrid (dark blue) categorization approaches. The results are statistically significant by the Friedman Test [FriedmanTest 2012].

These results suggest that directly categorizing bird images is difficult for

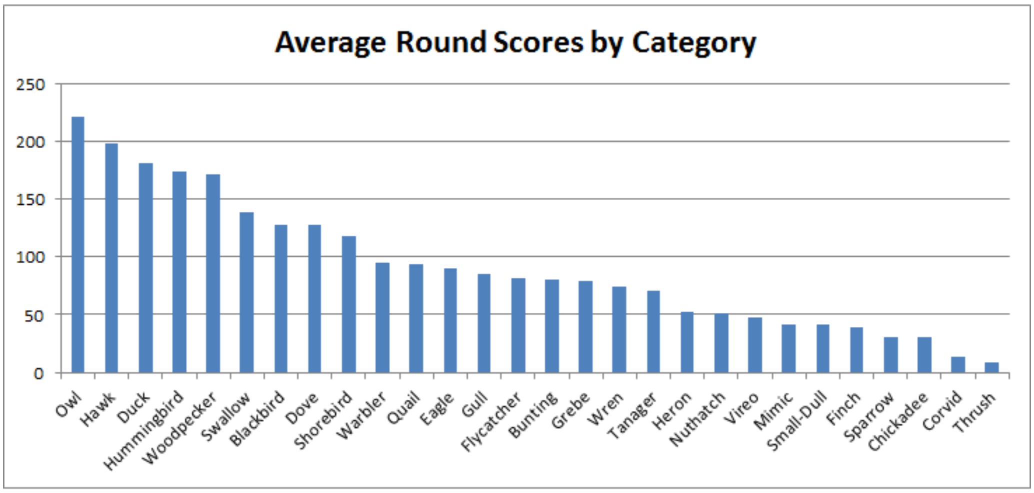
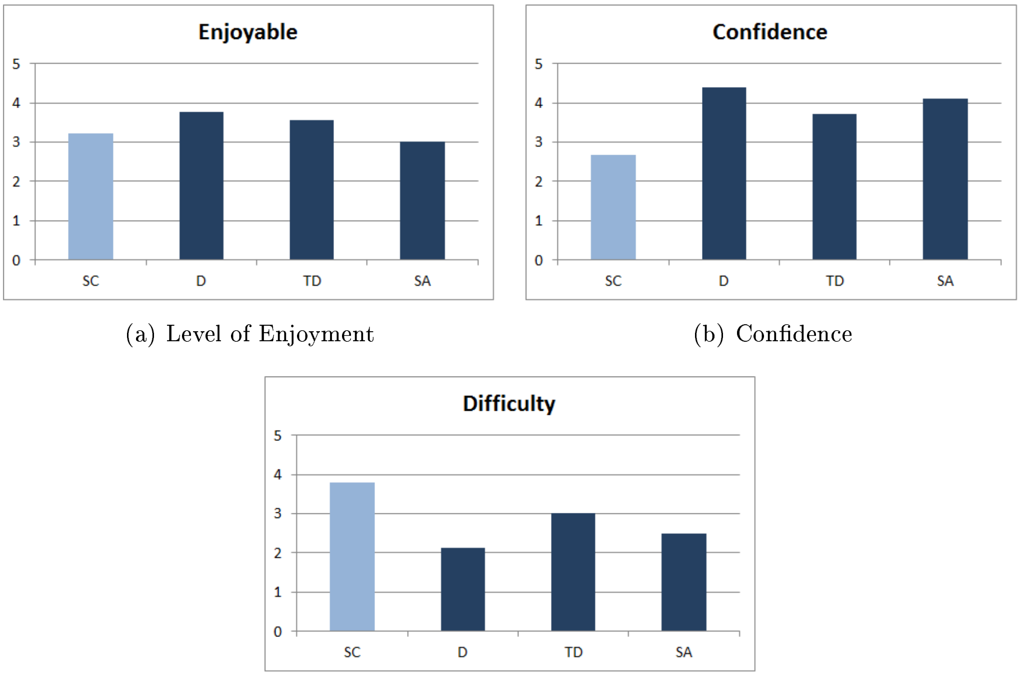


Figure 4.9: Average Round Score by Category

players. This is true especially for categories that are less commonly known. Figure 4.9 shows the average round score by family-level categories in the SC version of the game, showing that players were less successful at classifying bird images directly when the categories are unfamiliar.

The game statistics suggest that SC version (where players score categorical attributes) is the most difficult and the D version (where players name and score attributes of their own choosing) is the least difficult. These results are consistent with the self-reported answers provided in the surveys. In the survey, we asked participants to rate, on a 5-point scale, how enjoyable the game is (1=not at all, 5=very much), how difficult the game is (1=very easy, 5=very difficult), as well as how often they are confident about their answers (1=rarely, 5=very often). Figure 4.10(a), 4.10(b) and 4.10(c) shows that players report having the least confident playing the SC of the game ($p << 0.001$), as well as finding it the least enjoyable ($p = 0.014$) and the most difficult ($p << 0.001$); in contrast, they are the most confident playing D version of the game, and find it the most enjoyable and the least difficult.

We also asked participants to rate, on a 5-point scale (1=rarely, 5=frequently) how often they encountered different types of difficulties in each version of the game. Table 4.3 shows that in the D version, the difficulties players encountered were attributed to their game partner's inability to select the right images; while in SC version, the difficulties were attributed to the fact that players themselves did not know what the attribute meant and which images to select. It is also interesting to note that players enjoyed the D version of the game, where they get to both name and score the attributes of their own choosing, more than the SA version of the game, where they as asked to score the attributes that were named by previous players.



(a) Level of Enjoyment

(b) Confidence

(c) Perceived Difficulty

Figure 4.10: Self-Reported Game Experience

Types of Difficulties	SC	D	TD	SA
I have a hard time coming up with an attribute	NA	1.63	2.28	NA
I didn't know what the attribute meant	3.77	1.33	1.56	1.44
I didn't know which images to select	3.80	1.77	1.94	2.00
I selected the right images, but my partner didn't	2.67	2.27	2.06	2.11

Table 4.3: Frequency of different types of difficulties

All the results discussed thus far suggest that asking players to name and score perceptual attributes is better than asking them to score categorical attributes directly. Given that players neither enjoyed nor excelled at categorizing bird images directly, can we design a system that asks players to provide attributes of their own choosing, then uses machine learning algorithms to predict the actual categories? How would this hybrid categorization approach compare against the human categorization approach in terms of its performance (i.e., precision) in categorizing bird images? To answer this question, we trained a maximum entropy classifier using the attributes and attribute values collected from the D, TD and SA versions of the game to predict the actual bird categories.

Figure 4.11 shows the precision of both approaches across different coarse grained categories. The results are averaged over the 5-fold cross validation. Out of 28 categories, the hybrid categorization approach outperforms human categorization in 16 of the categories. It is interesting to note that some of the biggest performance gaps

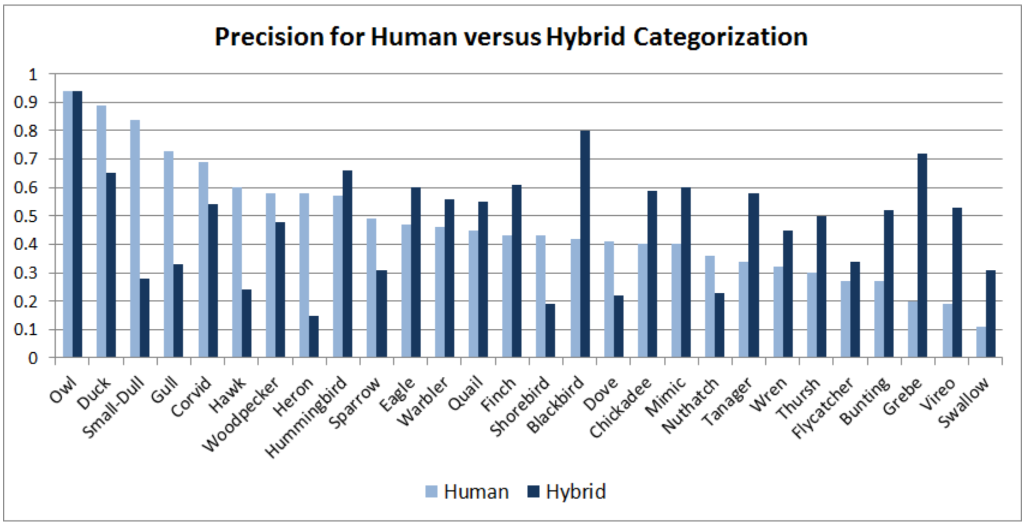


Figure 4.11: Precision: Human Categorization Approach versus Hybrid Categorization Approach

Category	Most Positively Weighted Attributes
Blackbird	black beaks, solid black tail, pointy thin tail, blu yelw bla head, blue head, yellow breast
Finch red	brown features, red head, yellow body, not yel/wh belly, long tail, red on body tail
Vireo	on thin branch, greenish head, not yellow belly, white belly, on branch, on branches, brownspotted breast, grey tail
Grebe	water, water wing, duck
Chickadee	not spotted wing, grey tail, odd white spot on heads, on branch, shorter beak
Mimic	whitish belly, not all wh breast, beige or tan breast, very long beak
Tanager	red and yello, red body, flat tail, red head, red feathers, red parts
Bunting	blue head, slanting up tail

Table 4.4: Attributes most positively weighted by the maximum entropy classifier

when humans perform better are categories that are more common, such as Dove, Woodpecker, Shorebird, Sparrow, Gull, Heron, Hawk, Duck; while the categories where the hybrid approach performs better are categories that are less visually familiar, such as Black Bird, Grebe, Chickadee, Mimic, Tanager, Bunting, Finch, etc. Table 4.4 shows the most positively weighted attributes in the maximum entropy classifier for these categories.

Note that the data used to train the maximum entropy classifier is extremely sparse, i.e., many attribute values are missing. Even so, the hybrid categorization approach outperforms human categorization, especially for predicting categories that are less familiar to humans. The performance of the hybrid categorization approach can be further enhanced by allowing more players to complete the matrix, then retraining the learning algorithm. In conclusion, the hybrid categorization approach is superior: it satisfies not only the *task-centric* objective of the system, but also the *human-centric* objective, since naming and scoring attributes of their own choosing is a much easier and enjoyable task for game players.

4.4 Case Study II: Noun Phrase Categorization

In the previous case study, we show that knowledge limitations can severely hinder human performance in a task where workers are unfamiliar with the attributes (e.g., they do not know how the birds belonging to the species "Corvids" look like), but that these limitations can be overcome with the help of machines. In this next case study, we consider a different kind of knowledge limitations, where human workers are familiar with the attributes but unfamiliar with the entities themselves. Specifically, we consider how to design a human computation system where humans verify categorical facts (e.g., "Washington is a city," "Marion Cotillard is an actress") that are extracted from a web mining system called NELL (Never-Ending Language Learner).

4.4.1 Motivation

NELL is a web mining system that continuously reads the Web and extracts categorical attributes (e.g., "Husky is a dog") and relational attributes (e.g., "Sidney Crosby plays for the Pittsburgh Penguins") of real world entities. NELL consists of four major components: (1) coupled pattern learner (CPL), which uses context patterns such as 'animals such as X", or 'X is headquartered in Y" to extract instances of categories and relations, (2) coupled SEAL (CSEAL), which mines lists and tables to extract instances of a certain category or relation, (3) coupled morphological classifier (CMC), which uses logistic regression to map morphological features (e.g., words, capitalization, affixes, part of speech) to categories, and (4) rule learner (RL), which learns probabilistic Horn clauses. The system starts with a manually crafted ontology (with a set of categories and relations and a set of constraints, e.g., mutual exclusion between certain categories) and a set of seed examples for each predicate in the ontology. Each of the four components proposes candidate facts and beliefs, along with probabilities and a summary of the evidence. A Knowledge Integrator then evaluates the reports from each component, and makes a decision as to whether to promote any given candidate facts.

It has been noted that iterative learning in NELL can suffer from the accumulation of errors if left unsupervised [Carlson 2010a]. For example, the class "baked goods" suffers from concept drift due to the incorrectly promoted named entity 'internet cookie", and becomes a category populated with entities related to computers instead of pastries. In addition, the NELL ontology of categories is manually specified by humans. It is not yet known how to compare different manually specified ontologies, and how they may or may not provide adequate coupling constraints [Carlson 2010b] for NELL. There has been recent efforts on extending the ontology automatically, by learning new subclasses (e.g., using text patterns like 'Y like cows and X," 'X and other nonhuman Y," 'X are mostly solitary Y," 'X and other hoofed Y") and relations [Mohamed 2011]. An important next step for NELL, therefore, is to consider what new thing to learn next [Banko 2007]. These observations all point to the need for human supervision in NELL. In this work, we address the challenges in designing a human computation system to elicit one form of human supervision, where the crowd helps to verify facts (i.e., categorize noun phrases) that are extracted by NELL.

NELL is also a *continuous* web mining system, built to extract massive amount of facts on a daily basis. To be cost-effective, we would ideally want a human computation system where people are motivated to *volunteer* their time and effort in verifying the facts in NELL. Designing a system that enables people to accurately categorize noun phrases is challenging: many of the noun phrases extracted by NELL refer to real-world entities that are obscure and not commonly known to people. It is even more challenging to embed such a task in a game, where players may be discouraged by their inability to complete the task successfully. In designing a human computation system for verifying facts in NELL, we tackle two questions. First, how severe is the knowledge limitation in noun phrase categorization? Can we enhance people's ability to categorize noun phrases, by transforming noun phrases automatically into another representation (i.e., images), and asking workers to classify the images instead? Second, can we transform the noun phrase categorization task into a game format? Through two Mechanical Turk experiments, we show that (i) the performance of human workers in categorizing noun phrases improves dramatically when they are provided with images than if they are given the noun phrase alone, and (ii) the complementary-agreement mechanism can be applied successfully to the noun phrase categorization task, suggesting that games with a purpose is feasible avenue for human supervision for NELL.

4.4.2 Knowledge Gap and Entity Representation

A central hypothesis in this case study is that transforming the representation of the entities (i.e., from noun phrases into images) will enhance the ability of the crowd to categorize noun phrases, despite the fact that the noun phrases extracted by NELL are not always commonly known by an average worker. To test this hypothesis, we conducted a Mechanical Turk study, where we asked workers to perform a series of multiple-choice categorization tasks. In each task, workers are shown a noun phrase and asked to classify the noun phrase into one of four categories. If none of the choices are appropriate, workers can choose "other" as the category and specify what he or she believes the category to be. In addition to selecting a category, workers must specify their confidence in their answer, by selecting one of four submit buttons ("just guess," "unsure," "quite sure," "100% certain.")

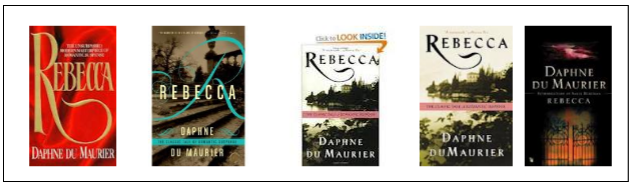
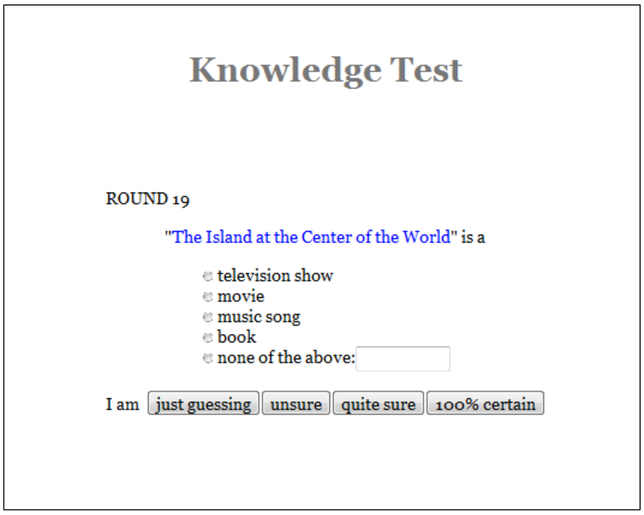


Figure 4.12: Google images extracted using "rebecca book" as the search query

The experiments have two conditions - the vanilla view condition, where workers are shown the noun phrase only (Figure 4.13), and the image condition, where workers are shown a noun phrase and a set of Google images (up to 8) associated with the noun phrase (Figure 4.14). The Google images are extracted from the Web automatically, using "*np category*" as the search query, where *category* is the current, highest probability category of the noun phrase in NELL (see Figure 4.12 for an example). Despite the danger of biasing workers, including *category* in the search query is necessary because there are many noun phrases that are either underspecified (e.g., "Rebecca") or polysemous (e.g., "Washington").



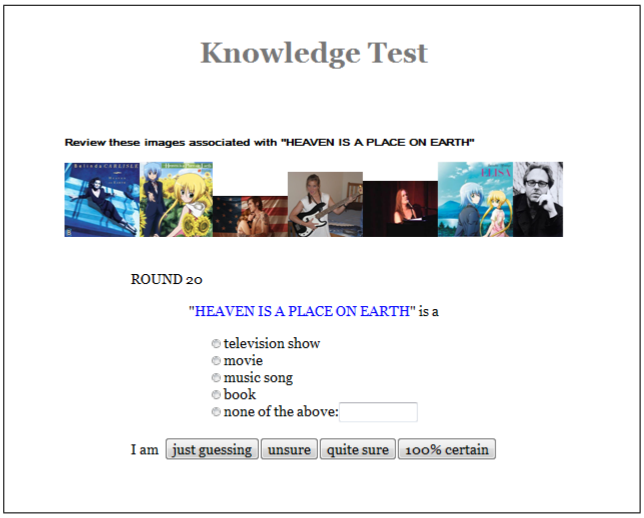
Knowledge Test

ROUND 19

"The Island at the Center of the World" is a

○ television show
○ movie
○ music song
○ book
○ none of the above:

Figure 4.13: Vanilla View



Knowledge Test

Review these images associated with "HEAVEN IS A PLACE ON EARTH"

ROUND 20

"HEAVEN IS A PLACE ON EARTH" is a

○ television show
○ movie
○ music song
○ book
○ none of the above:

Figure 4.14: Image View

The Mechanical Turk task provides workers with instructions to log onto a server and complete a series of 20 multiple choice questions. The system keeps track of which noun phrases each worker has seen, and serves him or her one, not both, of the views for any given noun phrase. There are 3919 noun phrases in total belonging to 20 different NELL categories. The noun phrases are randomly drawn from the head and tail list (in terms of occurrence frequency on the Web as measured by Google HIT count) of the noun phrases in each category. Each noun phrase is classified by at least 5 unique workers. In total, our system collected 21,615 classifications by 422 unique workers. Workers were paid 10 cents for their participation.

4.4.2.1 Results

Do workers face knowledge limitation in categorizing noun phrases? If so, how severe is this limitation? How does this limitation vary across different categories? We compare the vanilla view and image view using four different measures: response entropy, response confidence, agreement with NELL, and categorization performance (as measured against ground truth).

Response Entropy

For each multiple choice question, the response entropy R_e is a measure of the consensus level between the responses of independent workers: $R_e = -\sum_{c \in \mathcal{C}} p_c \cdot \log_2 \, p_c$, where $\mathcal{C}$ is the set of choices of categories offered by the question (including the category *other*). Figure 4.15 compares the average response entropy (over all the multiple choice questions) in the vanilla versus image view across 20 different categories. Results show that workers have lower response entropy (i.e., higher consensus) when given the image view than the vanilla view in all but one (i.e., the category *bank*) categories. The differences in response entropy between the vanilla and image views are statistically significant for every category by the paired t-test (p = 0.03), except for the categories *bank*, *biotech company*, *music song* and *movie*. This graph displays the categories in order of lowest to highest response entropy in the vanilla view, showing that the easiest noun phrases to categorize without images are *bank*, *winery*, *plant*, *movie*, while the hardest are *ceo*, *athlete*, *coach* and *music song*. For categories such as *bank* and *winery*, one explanation for the high level of consensus between workers is that the noun phrases themselves give the category away – e.g., since names of banks usually contain the word “bank” and names of wineries typically contain the word “winery.”

Figure 4.16 shows how much the noun phrase to image transformation actually helps to lower the response entropy (i.e., increase consensus). Results show that for many of the categories that are most challenging for humans, e.g., athlete, coach, book, writer and chef, the response entropy is reduced $\geq$ 38% when workers are shown the image view. Amongst the most difficult, there are some categories – e.g., ceo and music song – where the presence of images (i.e., $\leq$ 20% reduction) did not help much. In other words, while for some categories images help to reveal the identity and category of the named entities, for other categories the images are much less informative. In the extreme case, e.g., *bank*, images actually caused

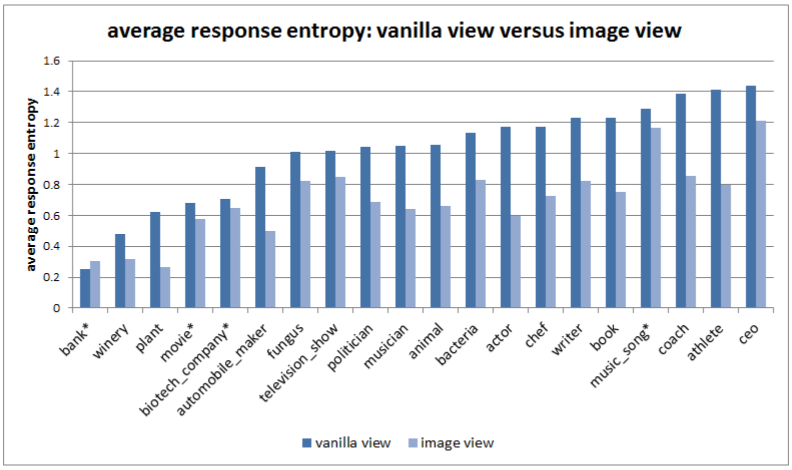


Figure 4.15: Average response entropy by category: vanilla versus image view. The differences are statistically significant in every category except the ones marked with *.

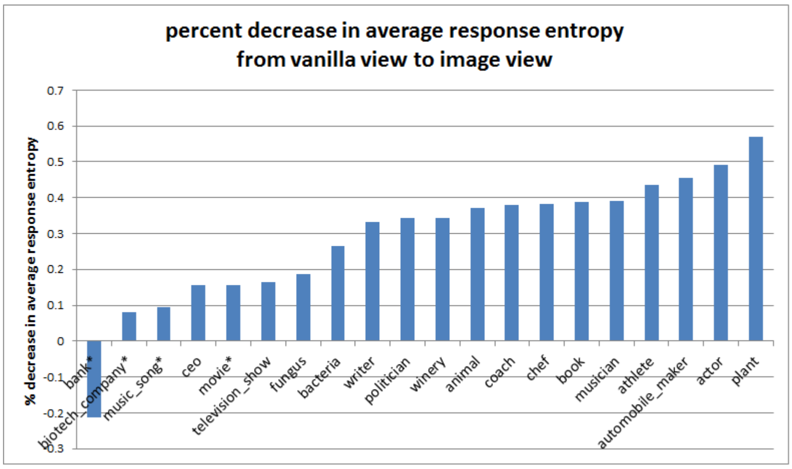


Figure 4.16: Percent decrease in response entropy going from vanilla view to image view

confusion, resulting in workers diverging in their answers.

Response Confidence

In terms of response confidence, the results are very similar. For each multiple choice question, the response confidence is 0 if the workers selected "just guessing" upon submitting their answer, 0.25 for "unsure," 0.75 for "quite sure," and 1 for "100% certain."

Figure 4.17 compares the average response confidence (over all the multiple choice questions) in the vanilla versus image view across 20 different categories. Results show that workers have higher response confidence when given the image view than the vanilla view. The differences in response confidence between the vanilla and image views are statistically significant for every category by the paired

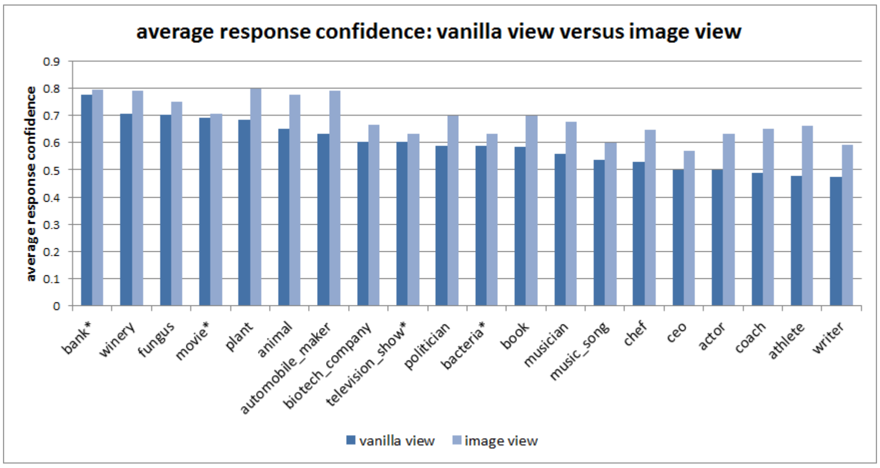


Figure 4.17: Average response confidence by category: vanilla versus image view. The differences are statistically significant in every category except the ones marked with *.

t-test (p = 0.02), except for the categories *bank*, *movie*, *television show* and *bacteria*. This graph displays the categories in order of highest to lowest response confidence in the vanilla view, showing that the easiest noun phrases to categorize without images are *bank*, *winery*, *fungus*, *movie*, while the hardest are *writer*, *athlete*, *coach* and *actor*. These results are fairly well aligned with the response entropy results: the categories in which workers have the least (or most) consensus with each other are also ones where they are the least (or most) confident in their individual responses.

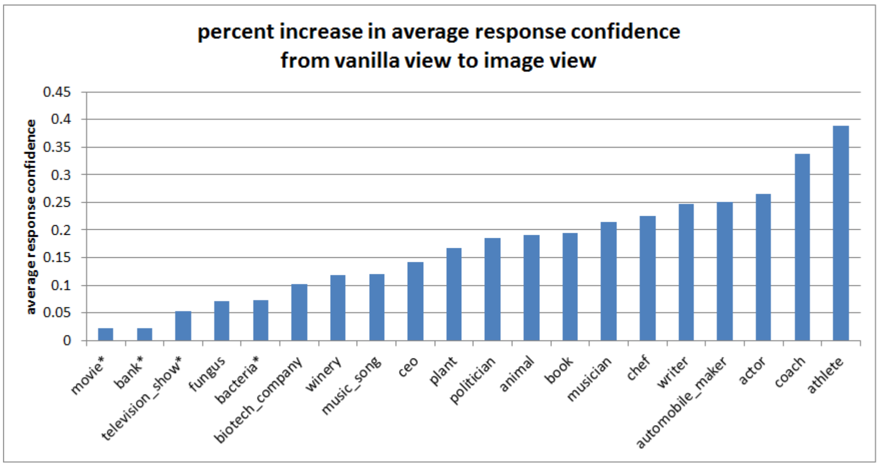


Figure 4.18: Percent increase in response confidence going from vanilla view to image view

Figure 4.18 shows how much the noun phrase to image transformation actually helps to increase the response confidence. Results show that for many of the categories where workers have the least confidence in their responses, e.g., writer, athlete, coach, actor, ceo, the response confidence is increased between 22% to close to 40% when workers are shown the image view. Likewise, there are some categories – e.g., *movie* and *bank* – where the improvement in confidence is minimal ($\leq 2\%$).

Agreement with NELL

Overall, the results so far suggest that by transforming noun phrases into images, workers converge more frequently towards the same answer *and* have higher confidence in their individual response. In addition to these effects, we also observe that the image view produced predictions that agree with NELL more often than the vanilla view.

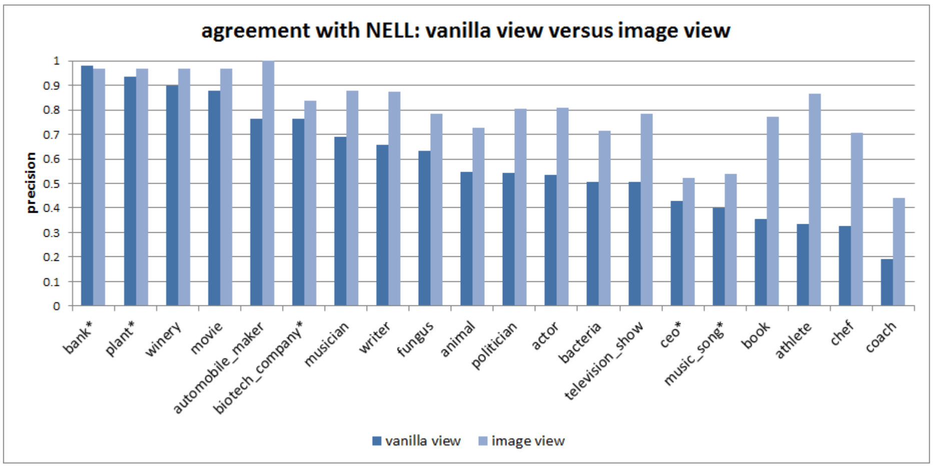


Figure 4.19: Agreement with NELL: vanilla versus image view. The differences are statistically significant in every category except the ones marked with *.

We evaluate the crowd's predicted category (by majority) for each noun phrase, using the highest probability category inferred by NELL as ground truth. The crowd's precision in a category, then, is measured by $tp/(tp+fp)$, where tp is the number of true positive predictions and fp is the number of false positive predictions. Results (Figure 4.19) show that in all but one category, the crowd's prediction agrees with NELL more when images are present. The differences in precision between the vanilla and image views are statistically significant for every category by the paired t-test ($p = 0.03$), except for the categories *bank*, *plant*, *biotech company*, *music song* and *ceo*. This graph displays the categories in order of highest to lowest agreement in the vanilla view.

Again, we can look at the ranking of the categories in terms of the percentage increase in agreement with NELL, showing how much noun phrase to image transformation bias workers towards NELL's beliefs in different categories. Figure 4.20 shows there is more than 100% increase in agreement with NELL for the four most difficult categories, namely, *chef*, *book*, *coach* and *athlete*. This is particular interesting because NELL's beliefs are derived from analyzing text data, while the crowd's predictions are based on images.

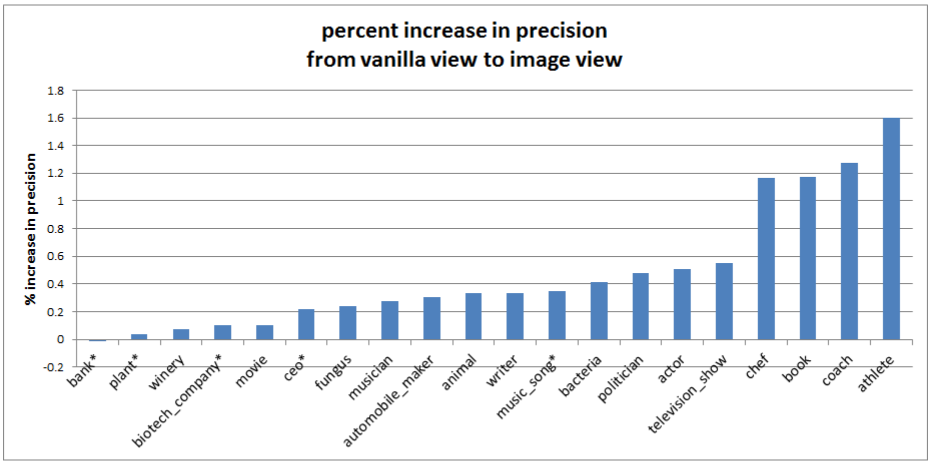


Figure 4.20: Percent increase in agreement with NELL going from vanilla view to image view

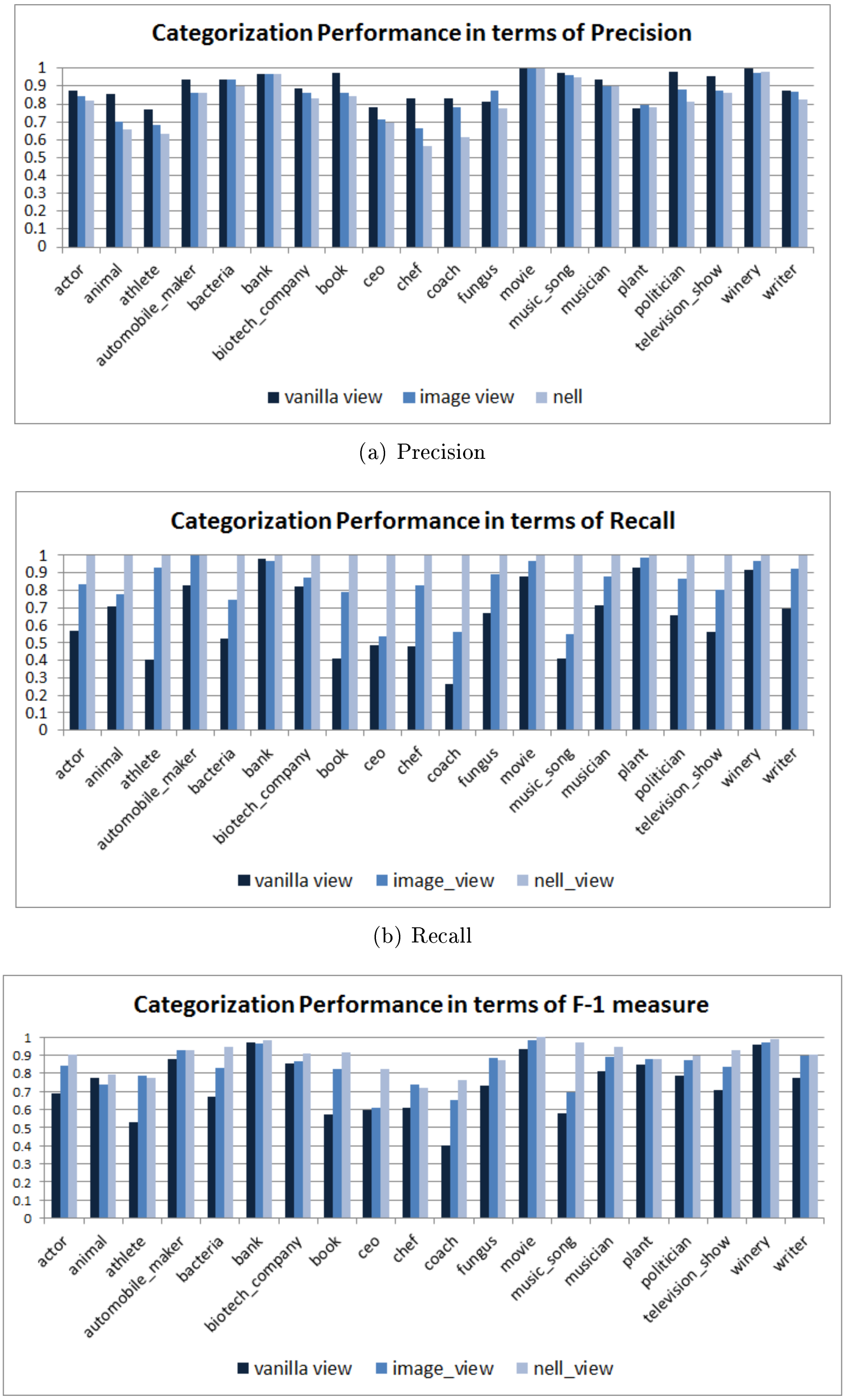
Categorization Performance

The ultimate question is whether this transformation *actually* helps in terms of resulting in more accurate categorizations. Note that using NELL's beliefs as ground truth is only an approximation of precision, as some of NELL's extractions might be incorrect (as mentioned in chapter 2, NELL is currently extracting millions of assertions, but with accuracy at roughly 85%). To get a better sense of the actual categorization performance of the vanilla view, image view and NELL, we obtained the ground truth category from Mechanical Turk, where we ask 3 workers to evaluate each (noun phrase, category) pairs, using search engines to check whether the noun phrase belongs to the category (if not, what the alternative category is), and also enter the URL of a webpage that contains evidence supporting their claims. We then use the majority vote category as the ground truth category of the noun phrase.

To answer this question, we evaluate the crowd's predictions in the vanilla view and image view as well as NELL's beliefs against the ground truth obtained from Mechanical Turk. Categorization performance can be measured in terms of precision, recall and F-1 measure. As shown in Figure 4.21 on the next page, results show that the vanilla view outperforms the image view and NELL in terms of precision, but is much worse in terms of recall. Overall, based on the F-1 measure, the image view and NELL outperforms the vanilla view in terms of categorizing noun phrases accurately.

4.4.3 Applying the Complementary-Agreement Mechanism

Categorizing noun phrases by images works because people appear to be familiar with the categories that are in NELL's ontology and what visual attributes map to those categories. Having established that humans can categorize noun phrase with higher consensus, confidence and accuracy when given images, next we will



(a) Precision

(b) Recall

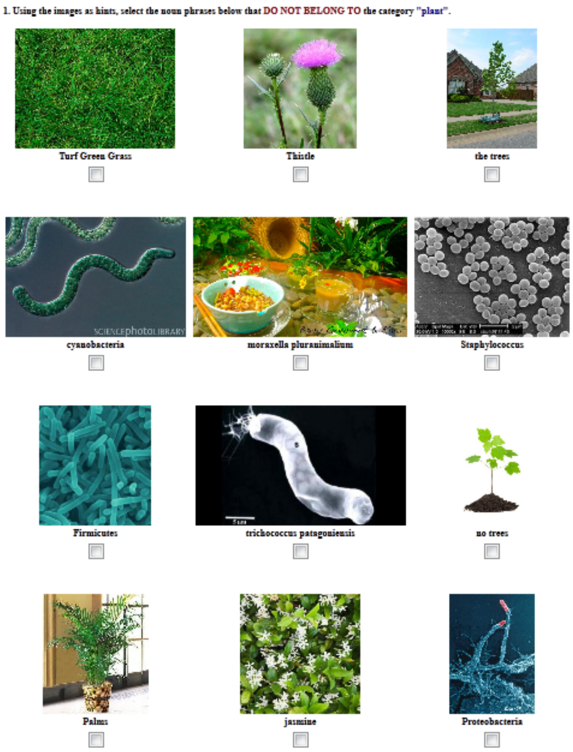
(c) F1

Figure 4.21: Categorization Performance: Vanilla View, Image View, NELL

investigate whether the noun phrase categorization task can be embedded in an image game like The Perfect Split, so that the crowd would be motivated to supervise a continuous, never-ending web mining system for free.

4.4.3.1 Experiments

We conducted a Mechanical Turk experiment, where workers are asked to perform the noun phrase categorization tasks using the complementary agreement mechanism. In each HIT, a worker is shown a category name (e.g., "plant"), a grid of 12 images each corresponding to a noun phrase, along with the noun phrases themselves beneath each image. We created the grid by choosing a category, then drawing some images that belong to that category, and others that do not belong to that category (according to NELL's beliefs). Note that each noun phrases is associated with between 3 and 8 images extracted automatically using Google Search; therefore, each noun phrase will appear multiple times as a different image in different grids.

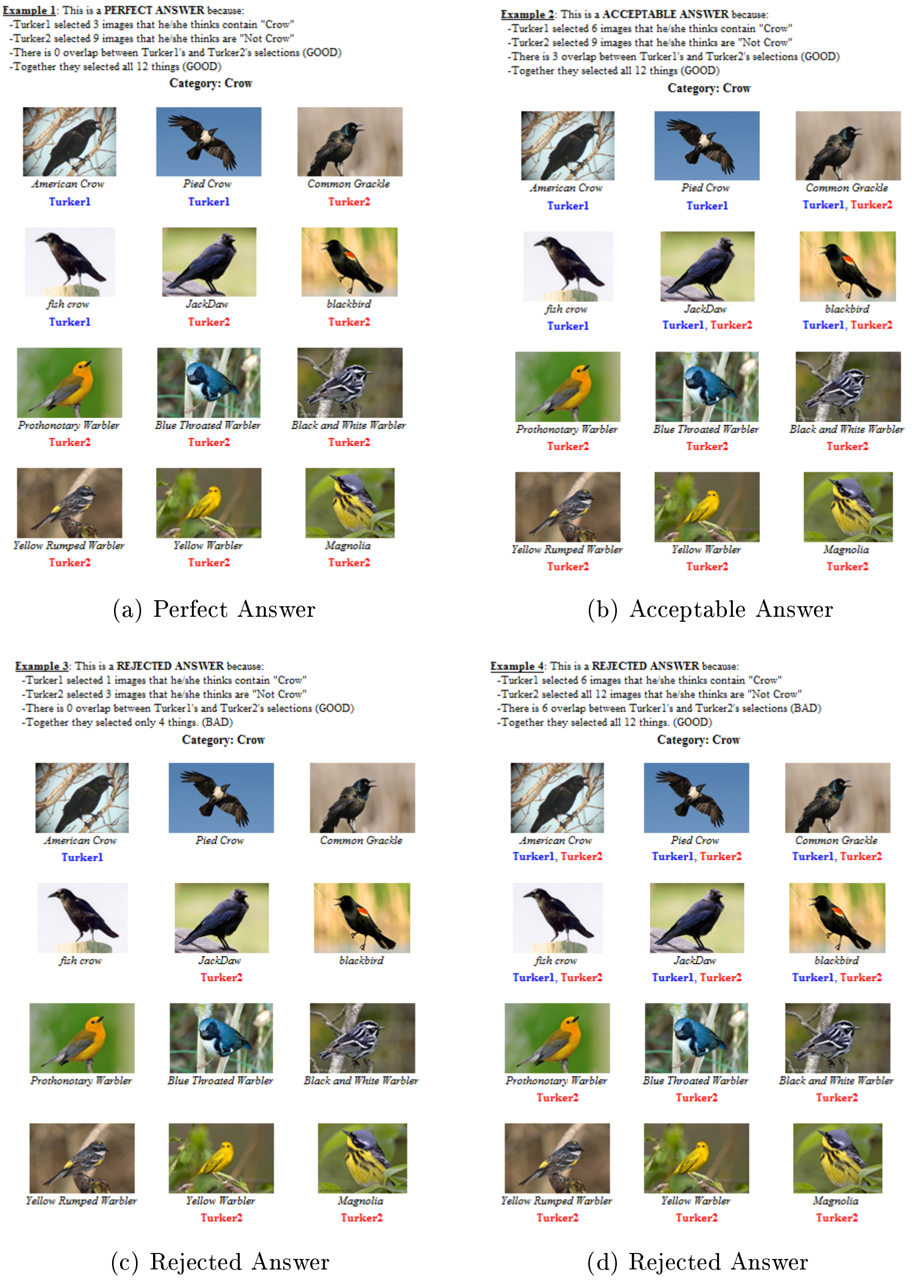


1. Using the images as hints, select the noun phrases below that DO NOT BELONG TO the category "plant".

Turf Green Grass ☐
Thistle ☐
the trees ☐
cyanobacteria ☐
moraxella pluranimalium ☐
Staphylococcus ☐
Firmicutes ☐
trichococcus patagoniensis ☐
no trees ☐
Palms ☐
jasmine ☐
Proteobacteria ☐

Figure 4.22: Complementary-Agreement Mechanism on Mechanical Turk: Example 1

For each noun phrase, there is a pair of HITs (Human Intelligence Task) to be completed by two different workers – one HIT will ask the worker to choose the images that *belong* to the category, the other HIT will ask the worker to choose the images that *do not belong* to the category. The pair of HITs are assignments that belong to two different Mechanical Turk tasks, labeled Version 1 and Version 2. Workers are told that they should perform HITs that belong to one of the versions



Example 1: This is a **PERFECT ANSWER** because:
-Turker1 selected 3 images that he/she thinks contain "Crow"
-Turker2 selected 9 images that he/she thinks are "Not Crow"
-There is 0 overlap between Turker1's and Turker2's selections (GOOD)
-Together they selected all 12 things (GOOD)

Category: Crow

(a) Perfect Answer

Example 2: This is a **ACCEPTABLE ANSWER** because:
-Turker1 selected 6 images that he/she thinks contain "Crow"
-Turker2 selected 9 images that he/she thinks are "Not Crow"
-There is 3 overlap between Turker1's and Turker2's selections (GOOD)
-Together they selected all 12 things (GOOD)

Category: Crow

(b) Acceptable Answer

Example 3: This is a **REJECTED ANSWER** because:
-Turker1 selected 1 images that he/she thinks contain "Crow"
-Turker2 selected 3 images that he/she thinks are "Not Crow"
-There is 0 overlap between Turker1's and Turker2's selections (GOOD)
-Together they selected only 4 things. (BAD)

Category: Crow

(c) Rejected Answer

Example 4: This is a **REJECTED ANSWER** because:
-Turker1 selected 6 images that he/she thinks contain "Crow"
-Turker2 selected all 12 images that he/she thinks are "Not Crow"
-There is 6 overlap between Turker1's and Turker2's selections (BAD)
-Together they selected all 12 things. (GOOD)

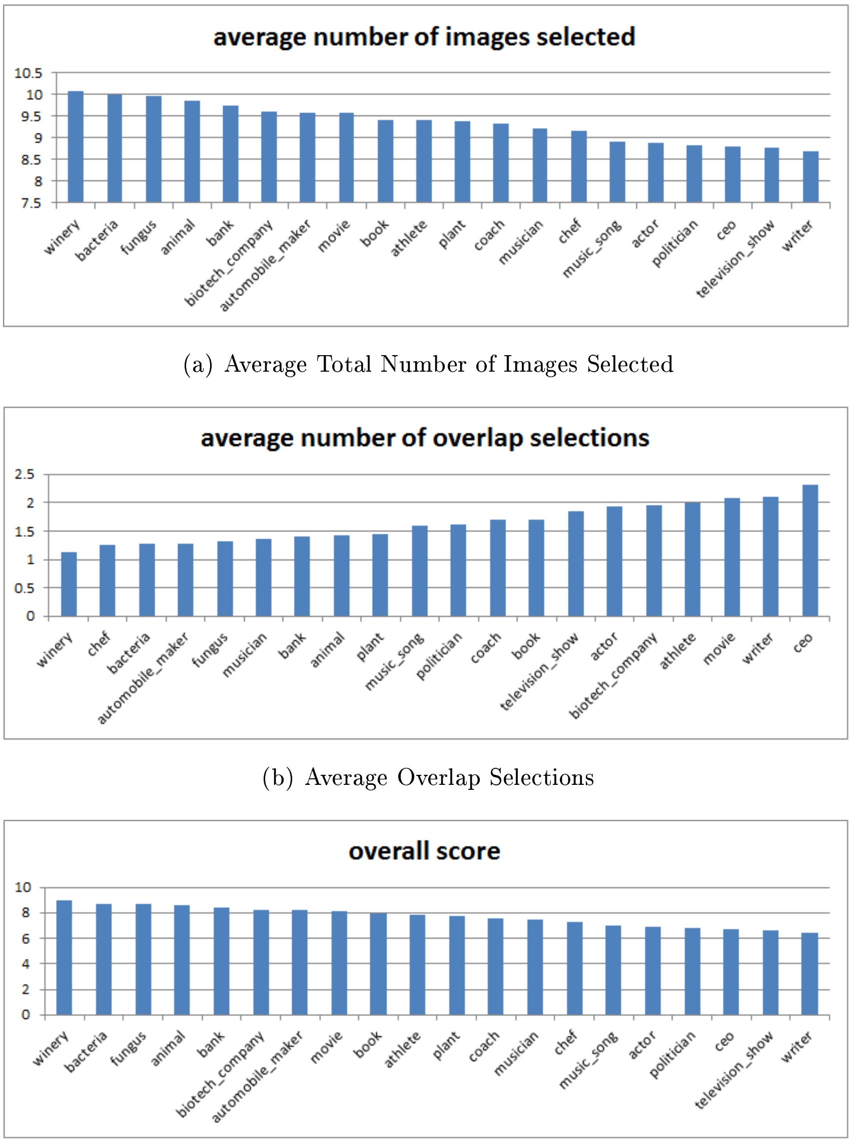
Category: Crow

(d) Rejected Answer

Figure 4.23: Tutorial of the Complementary Agreement Mechanism

only, and not both. There are 2650 HITs in each version, and a total of 1932 noun phrases to be verified using the game mechanism.

In the instructions, workers are told that their answer will be verified against that of another worker, and that (i) both he and the other worker must select at least 1 noun phrase, or both of their work will be rejected; (2) in the perfect situation, his selections and the other worker's selections do not overlap *and* together they should



(a) Average Total Number of Images Selected

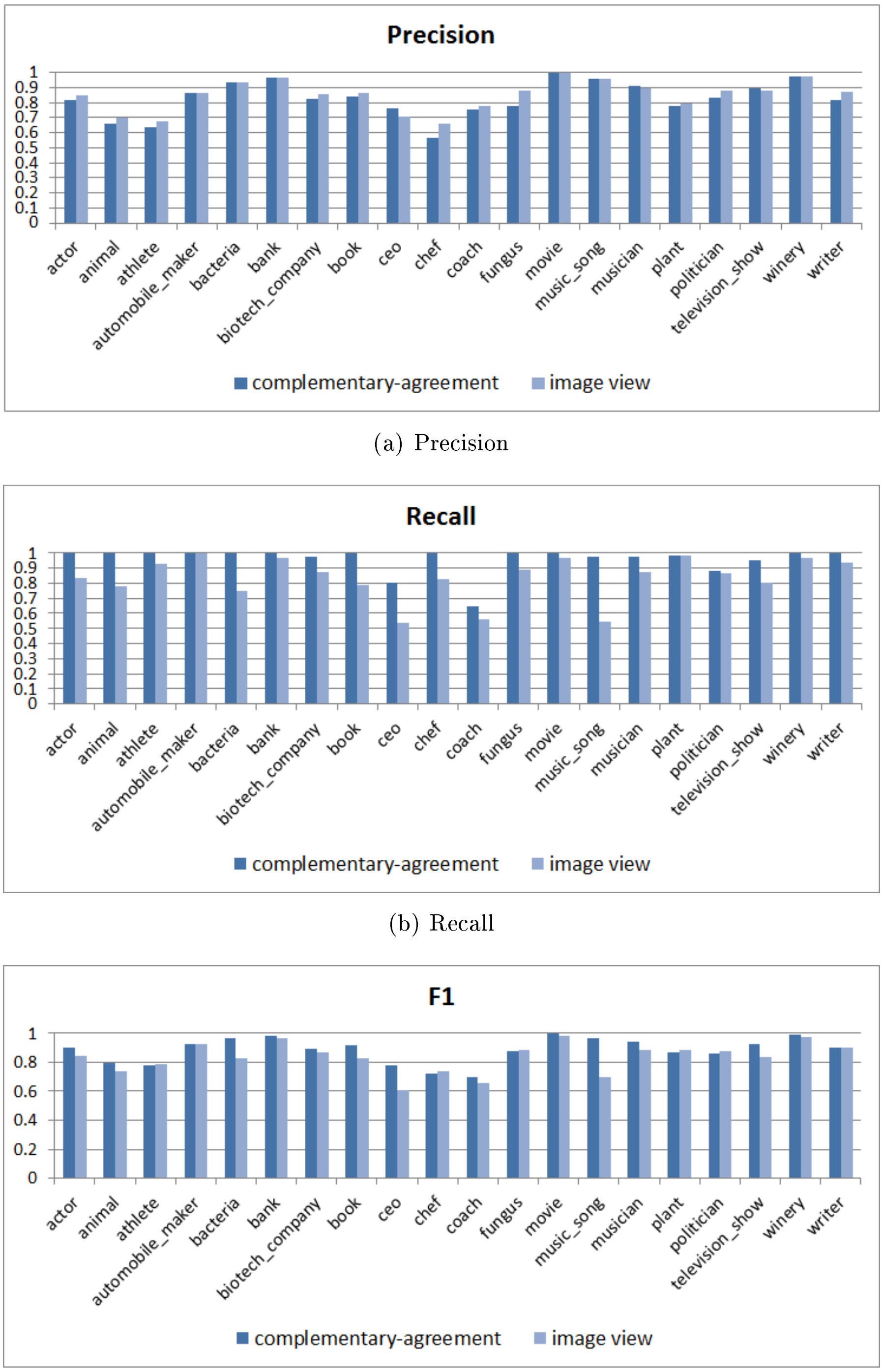
(b) Average Overlap Selections

(c) Overall Score

Figure 4.24: Human-Centric Objective

have selected all 12 noun phrases; and (3) if his selections and the other worker's selections overlap too much (> 3 noun phrases) if together they selected too few noun phrases (< 9), both of their work will be rejected. Workers are also given a tutorial (Figure 4.23), which shows examples of work that are considered the perfect answer, acceptable answer, and rejected answers.

Figure 4.24 shows how well the workers are able to perform the categorization tasks using the complementary-agreement mechanism across different categories, in terms of the number of overlap selections (Figure 4.24(b)), total number of images selected (Figure 4.24(a)), overall score (total number of images selected *minus* the



(a) Precision

(b) Recall

(c) F1

Figure 4.25: Categorization Performance: complementary-agreement versus image view

number of overlap selections) averaged over pairs of workers. Based on the overall scores, the easiest categories are *winery*, *bacteria*, *fungus*, *animal* and *bank*, and the most difficult categories are *writer*, *television show*, *ceo*, *politician* and *actor*.

Lastly, we evaluate how the performance of the workers compare when they categorize noun phrases using the complementary-agreement mechanism versus not using the complementary-agreement mechanism (i.e., by answering multiple choice questions under the image view). Results (Figure 4.25) show that across all the categories, categorization by complementary-agreement mechanism has slightly lower precision than the image view, but much higher recall. Based on the F-1 measure, complementary-agreement mechanism outperforms the image view in terms of categorizing noun phrases accurately.

The exact reason for this superior performance is unclear. There are many possible reasons: (1) in the complementary-agreement mechanism, each noun phrase is essentially evaluated multiple times using different grids, whereas in the image view, each noun phrase is evaluated five times but the same information is given to each worker; (2) in the complementary-agreement mechanism, each image of the noun phrase is *explicitly marked* as belonging to a category or not, whereas under the image view, workers are given all the images belonging to a noun phrase, and make a decision based on the entire set; (3) in the complementary-agreement mechanism, workers are making binary decisions, whereas in the image view, workers are given multiple-choice questions. Uncovering the real reason behind the superior performance of the complementary-agreement mechanism requires additional investigation in future work.

In this section, we address the challenge of engaging the crowd to evaluate facts from a web mining system, where workers' performance is severely hindered by their knowledge limitations. Our results show that workers are much more confident and capable of categorizing noun phrases via images. Furthermore, we show that asking workers to categorize noun phrases using the complementary-agreement mechanism yields better categorization performance (according to the F-1 measure) than asking workers to categorize noun phrases via images in a multiple-choice format, suggesting that human supervision of NELL via a game is not only feasible, but potentially more accurate.

4.5 Conclusion

4.5.1 Reference to the Framework

Similar to TagATune and the ESP Game, The Perfect Split (and more generally, the complementary-agreement mechanism) is a query model for completing the cells of the matrix.

Figure 4.26 and 4.27 shows an example of this query model: the system chooses a set of entities $\{e_i\}$ (e.g., images) to present to the players. In the scoring mode (Figure 4.26), the game selects an already revealed column of the matrix and presents the associated attribute a_j (e.g., "colorful") along with the set of entities to the players. In return, the game receives a set of attribute values $\{(a_j, v_{i,j})\}$, where $v_{i,j}$ is the number of times the attribute a_j has been associated with entity e_i by

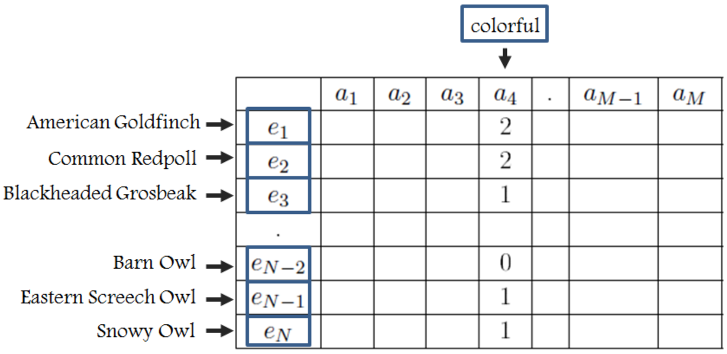


Figure 4.26: Query Model for The Perfect Split: Scoring Mode

.

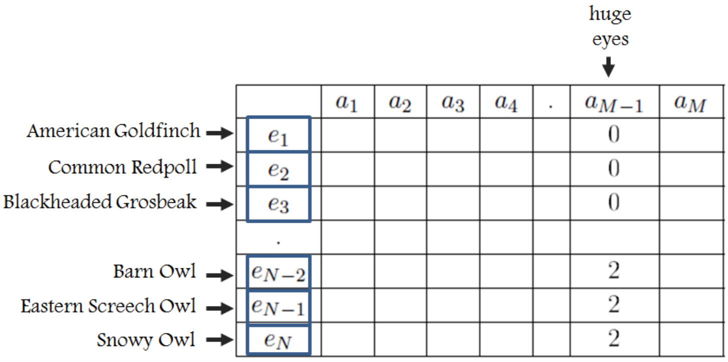


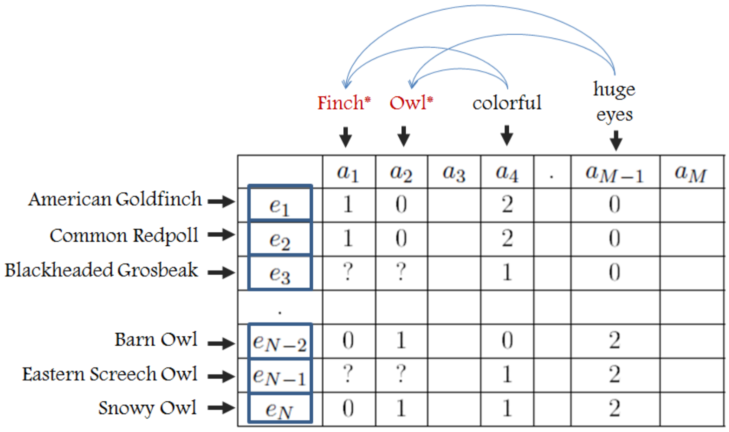
Figure 4.27: Query Model for The Perfect Split: Discovery Mode

.

different players. Note that in the example the counts returned by the two players can be 0 (both players agree that the attribute does not apply), 1 (only one of the players agree that the attribute applies) and 2 (both players agree that the attribute applies). In the discovery mode (Figure 4.27), the game presents only the set of entities to the player, and allows one of the players to reveal the column and then asks both players to fill in the attribute value for the entities.

In this chapter, we focus on the task of classifying attributes that are unfamiliar (e.g., bird species names) to players. Consider the example shown in Figure 4.28, where the attributes of interests are "Finch" and "Owl." Our results show that, instead of asking players to score unfamiliar attributes, a better approach is to query them for attributes of their own choosing (e.g., "colorful," "huge eyes"), then use the collected data as features to train a classifier to predict the attribute values of interests. This approach allows the game to satisfy both the *human-centric* and *task-centric* objectives of the system simultaneously.

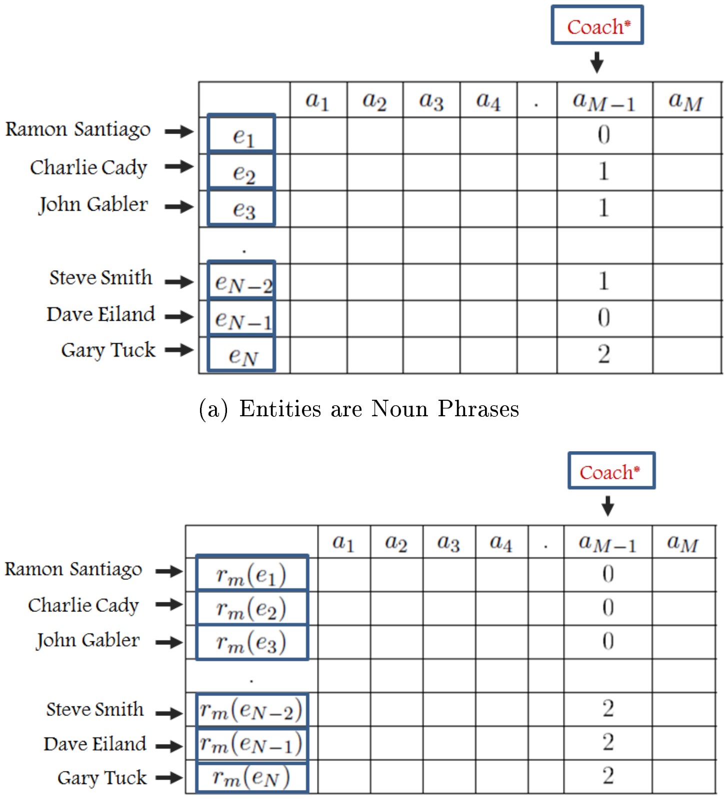
We also consider another type of knowledge limitation in attribute learning, namely, when workers are familiar with the attributes, but do not what attributes the entities have. Our case study focuses on designing a human computation system (ideally, a game that can collect data for free) for verifying categorical facts



		a_1	a_2	a_3	a_4	.	a_{M-1}	a_M
American Goldfinch	e_1	1	0		2		0	
Common Redpoll	e_2	1	0		2		0	
Blackheaded Grosbeak	e_3	?	?		1		0	
	.							
Barn Owl	e_{N-2}	0	1		0		2	
Eastern Screech Owl	e_{N-1}	?	?		1		2	
Snowy Owl	e_N	0	1		1		2	

Figure 4.28: Hybrid Categorization Approach

.



		a_1	a_2	a_3	a_4	.	a_{M-1}	a_M
Ramon Santiago	e_1						0	
Charlie Cady	e_2						1	
John Gabler	e_3						1	
	.							
Steve Smith	e_{N-2}						1	
Dave Eiland	e_{N-1}						0	
Gary Tuck	e_N						2	

(a) Entities are Noun Phrases

		a_1	a_2	a_3	a_4	.	a_{M-1}	a_M
Ramon Santiago	$r_m(e_1)$						0	
Charlie Cady	$r_m(e_2)$						0	
John Gabler	$r_m(e_3)$						0	
	.							
Steve Smith	$r_m(e_{N-2})$						2	
Dave Eiland	$r_m(e_{N-1})$						2	
Gary Tuck	$r_m(e_N)$						2	

(b) Entities are Noun Phrases Transformed into Images

Figure 4.29: Transforming the Representation of the Entities

extracted by a web-mining system called NELL. Our solution, as depicted in Figure 4.29 is to imperfectly transform the noun phrases into images, i.e., find a new representation $r^h(e)$ such that the output of the human computers is improved. Experiments show that workers can indeed categorize noun phrases with much higher consensus, confidence and accuracy if the noun phrases are transformed into images. Furthermore, the transformation allows the noun phrase categorization task to be embedded in a game setting, i.e., use the complementary-agreement mechanism to query for the values of the relevant columns of the matrix.

4.5.2 Lessons Learned

In this chapter, we introduce new techniques for enabling crowdworkers to perform knowledge intensive tasks, specifically, the task of scoring attributes when workers are unfamiliar with either the attributes or the entities themselves. Furthermore, we present a new, incentive-compatible game mechanism called complementary-agreement for handling attribute learning tasks both in the casual game and paid crowdsourcing settings, and evaluate its effectiveness in two contrasting case studies, namely bird image classification and noun phrase categorization.

In the case of bird image classification, we created a new game called The Perfect Split based on the novel complementary-agreement mechanism. In a user study involving 30 users, we find that game players struggle with the game when they are asked directly to categorize images of birds (especially those belonging to less commonly known species), both in terms of their game score as well as their ability to identify the correct species in the images; more importantly, players do not find the game appealing, which violates the *human-centric* objective of our human computation system. To address this problem, we introduce a hybrid approach for categorization – we ask players to provide attributes and attribute values of some perceptual characteristics (e.g., has blue head, has long tail) of the birds in the images, then use this player-generated data as features to train a simple maximum entropy algorithm to classify bird images into categories. Results show that the hybrid categorization approach outperforms the approach of asking players to categorize bird images directly. Furthermore, players have greater success at playing the game and rate the game as more enjoyable when they are asked to name and score attributes of their own choosing.

In bird image classification, game players have difficulty classifying images because they are unfamiliar with the attributes. In the second case study, we consider the situation where human workers are familiar with the attributes, but are unfamiliar with the entities themselves. Specifically, we address the following question: “how do we design a system where humans can help to categorize arbitrary noun phrases extracted from the Web by a continuous web-mining system, where many of the noun phrases are obscure and unfamiliar to them?” Similar to the case of bird classification, we find that human workers perform poorly at the categorization task due to knowledge limitations, but that this knowledge limitations can be overcome by transforming noun phrases into images and asking people to categorize noun phrases using the images as hints. Furthermore, we evaluate the use of the complementary-agreement mechanism to categorize noun phrases on Mechanical Turk, by presenting images and noun phrases in a grid, and asking workers to find complementary sets of entities that belong or do not belong a particular category. Results are promising – they show that workers are able to achieve low (less than 3 out of 12 images) overlap 83% of the time. In addition, workers have higher categorization performance when asked to categorize noun phrases using the complementary-agreement mechanism than to answer multiple-choice questions directly. This suggests that a game like The Perfect Split can be used to elicit the help of game players to supervise NELL, though more experimentation is required to confirm this speculation.

In both cases, we take the approach of designing a system to achieve the *human-centric* objective of the system, then using machines to either overcome some limitations of the human worker or complement their ability. Results show that our hybrid human and machine computation approach allows knowledge-intensive tasks to be completed more successfully than using human computation alone.

Chapter 5

Design Space of Human Computation Games

While there has been a proliferation of human computation games in the past few years, there has been little work on defining the design space of human computation games. In this section, we will outline the design space of games for attribute learning, including questions such as: What are the main components of a human computation game for attribute learning? What are the design decisions important for each of these dimensions? How can the existing game mechanisms – output-agreement (e.g., ESP Game), problem inversion (e.g., Verbosity, Phetch), input-agreement (e.g., TagATune), complementary-agreement (e.g., The Perfect Split) – be described or classified in relation to the design space? How does one know what game mechanism to apply and when?

5.1 Components and Design Decisions

For attribute learning, human computation games are essentially platforms that serve tasks for scoring or discovering attributes for a given set of entities, and reward players when the tasks are completed successfully. Therefore, this type of human computation games has five main components: (i) input, (ii) query, (iii) output, (iv) mechanism, (v) player.

The first two components – input and query – determine *what question* the game is asking the players in each round, and *which* input entities to focus on. Design decisions related to these two components include the following: In each round of the game, do we ask players to name relevant attributes, score attributes, or both? Are players given a single entity, or multiple entities to process? Depending on the answers to these questions, the game is essentially designed to ask different types of attribute discovery and scoring questions. For example, in the output-agreement mechanism (e.g., the ESP Game), players are presented with a single entity (e.g., an image) and asked to provide a tag. The ESP game is essentially asking a **single-entity, open-ended question**. In the problem-inversion mechanism (e.g., Verbosity and Phetch), the game presents a single entity to the describer (e.g., a word in the case of Verbosity, and an image in the case of Phetch) and asks him to provide a tag to the guesser. This game is also asking the players, or more specifically the describer, to answer a single-entity, open-ended question. Likewise, in TagATune, each player is given a music clip and asked to generate tags and share them with his partner; in other words, each player is essentially answering a single-entity, open-ended question.

In contrast, in The Perfect Split, the game presents a set of entities to both players. In the discovery mode, the game asks the positive player to name a relevant attribute, i.e., answer a **multiple-entities, open-ended question**. In the scoring mode, the players are given an attribute and asked whether that attribute applies to each entity in the set, i.e., answer a **multiple-entities, close-ended (binary-choice) question**. Open-ended questions allow players freedom to enter any attributes, which has the effect of making the game more fun but the data more noisy. Closed-ended questions give players less freedom, but allow attributes to be collected in a much more targeted way.

The third component pertains to whether there is any explicit restrictions on what players can enter into the system. Both TagATune and Phetch, for example, impose zero restrictions on what the players can enter. In contrast, the ESP Game prevents players from entering some tags (called taboo words) that have been frequently entered by previous players. In Verbosity, the describer must describe the attribute of the secret word/object using one of the sentence templates (e.g., "It is [blank]," "It has [blank]") provided by the game. In The Perfect Split, players are asked to provide an attribute that is postfixed by the name of a body part (e.g., head, neck, tail, wing) provided by the game.

The fourth component – mechanism – is a major component that distinguishes human computation games from other paid crowdsourcing platforms. A mechanism consists of a set of rules that determines how the players should interact with each other or with the system, as well as a reward system that determines when players are successful and how much they should be rewarded. This set of rules determine the answers to questions like "what information are the players allowed to communicate with the system or with each other?", "is the communication between players, if any, uni-directional or bi-directional," or "are all the players given the same set of entities, or are certain entities hidden from some players?"

Different from the typical reward system used in paid crowdsourcing platforms (e.g., Mechanical Turk) where workers are paid using a flat rate, human computation games reward workers when they are successful in achieving a goal, where the goal is designed to be always easier to reach if and only if workers are telling the truth. For example, in the ESP Game, players are rewarded only if their tags match; since communication between game partners are prohibited and the image is the only information that both players share, players are more likely to succeed in agreeing with each other if they make use of the shared information and enter tags that describe the content of the image. In TagATune, players are more likely to be able to guess whether the two pieces of music are the same or different, if they describe to their partners as accurately as possible what they are listening to. In problem-inversion games (e.g., Verbosity), the guesser is more likely to succeed in guessing the secret entity if the describer enters accurate tags that singly describe that entity and distinguish it from other entities. Lastly, in The Perfect Split, players are less likely to overlap with each other if they are careful to name an attribute whose values are easy to determine from the images, and if they select *only* the images that have (or do not have) a particular attribute, no more and no less. The last three games are distinctly different from the ESP Game in that communication between game partners are allowed, and the mechanism is designed to encourage truthful communication between game partners.

Finally, there are a set of design decisions about the players in the game, including the number of players allowed in each game (e.g., a pair versus multiple) as well as how the players are matched with each other (e.g., randomly versus by some criteria, such as expertise or compatibility). Most game mechanisms to-date involve only two players, with the exception of Phetch, where there is one describer but many guessers all racing to be the one who guesses the secret image first. In general, games that involve multiple players lend themselves well to competitive mechanisms, which have been much less studied than cooperative game mechanisms.

1. input
How many entities are presented to the players?
How are these entities chosen?
2. query
Are players asked to answer open-ended or closed-ended questions?
Are players asked to name relevant attributes, score attributes, or both?
3. output
Are there any restrictions on what players can enter as their outputs?
4. mechanism
What are the rules of the game?
Are communication between players allowed?
Is the communication uni-directional or bi-directional?
Are all the entities visible to all players, or are some entities hidden from some players?
How does the game determine when players have successfully completed the task?
How much are players rewarded?
Are the players supposed to cooperate or compete?
5. player
How many players are allowed in each game?
How are the players matched (e.g., randomly, by expertise/compatibility)

Table 5.1: Design Space

Table 5.1 summarizes the list of components of human computation games for attribute learning, and the design decisions that are associated with each of these components.

5.2 Choosing a Game Mechanism

Given the design space described above, how does one decide which game mechanism to use when? Table 5.2 highlights some of the key differences amongst existing games in the context of the design space. Based on this table, we will discuss a few useful questions to consider when choosing a game mechanism to apply to a given attribute learning problem.

First, do the entities of interests have high description entropy, i.e., is there a diverse language for describing attributes associated with the entities? For example, in music, the same mood can be described using many different words; while for

	ESP Game	**Verbosity**	**Phetch**	**TagATune**	**The Perfect Split**
input	single	single	single	pair	multiple
query	open-ended	open-ended	open-ended	open-ended	open-ended and close-ended
output restriction	taboo words	template	none	none	template
success criteria	agreement	function computation	function computation	function computation	complementary agreement
nb of players	2	2	multiple	2	2

Table 5.2: Games in the Design Space

images, objects can be described using more restricted language. In the case of vocabulary-rich input objects, a suitable game mechanism would be one that allows communication between game partners and does not enforce that players agree with each other. TagATune, Verbosity and Phetch are all examples of this scenario. In Verbosity, there can be a large number of attributes associated with the words that refer to real-world entities, e.g., *apple* can be described by many different attributes (including color, function, taste, category, etc.) and the number of ways to describe the same attribute can be varied (e.g., “It is red,” “It has red skin,” “It looks red.”) Likewise, by allowing communication, the game Phetch achieves the intended goal of collecting elaborate descriptions of images, useful for improving accessibility for blind users on the Web. In both cases, agreement-based mechanisms would not be appropriate.

The second question to consider is the actual task-centric objective of the system. For example, if the goal is to extract attributes such that each entity is distinguished from each other, then the game can present pairs of entities to the players and ask them for attributes that distinguish them, as in TagATune. If the task is to discriminate between sets of entities, then the complementary-agreement mechanism would be appropriate, as it presents a *set* of entities to players and asks for an attribute that are common to some entities and not others. In general, game mechanisms that serve only a single entity are unlikely to yield attributes that are discriminative.

Finally, The Perfect Split (which implements the complementary-agreement mechanism) is the only game (amongst the ones shown in Table 5.2) that asks close-ended questions, i.e., it presents an attribute and a set of entities to players and asks players to specify the value of that particular attribute for those entities. Being able to *explicitly* ask for the value of an attribute opens up new opportunities for hybrid human and machine computation – the machine can automatically discover a new attribute and its values for all the entities, then ask game players to evaluate its performance via a game like The Perfect Split. In fact, we have illustrated this phenomenon in the case study using The Perfect Split game to evaluate facts extracted by a continuous web mining system called NELL.

Chapter 6

Conclusion

6.1 Summary

The goal of this work is to advance our understanding about the design of hybrid human and machine computation systems. In particular, we focus on the role of machines to complement human ability and overcome their limitations, instead of their role in optimizing the process of human computation. Within the context of the attribute learning problem, this thesis makes a set of conceptual and practical contributions as follows.

6.1.1 Conceptual Contributions

We introduce a framework for characterizing the problem of attribute learning. The attribute learning problem is framed as the problem of filling in a matrix, where the rows are entities and columns are all the possible attributes in the world. Given an objective (e.g., to distinguish between different entities or group of entities), the tasks of a hybrid human and machine computation system is to identify the relevant attributes (i.e., columns of this matrix) and to fill in the attribute values (i.e., complete the cells of the relevant columns) in this matrix.

Throughout this thesis, we have tackled a variety of questions within this framework:

1. We show that games with a purpose can be viewed as a query model for editing the matrix, and introduce two game mechanisms which can motivate workers to reveal columns of the matrix in order to discriminate between individual entities or groups of entities. For example, in chapter 3, we address the problem of learning attributes in vocabulary-rich settings, i.e., where the input object (e.g., music) can be described by a vast and diverse vocabulary. We introduce a new game mechanism called *input-agreement* (and a game called TagATune) for annotating music, which is capable of yielding music attributes that are much more discriminative than what the widely used output-agreement mechanism can collect. In this mechanism, two workers (or game players) are given either the same entity or two different entities, and are asked to exchange tags with each other until they are able to guess whether the entities given to them are the same or different. The input-agreement mechanism is a specific instance of a general category of game mechanisms called *function computation mechanisms*, where workers are each given partial inputs and must cooperate (e.g., by combining their partial inputs) in order to successfully compute an

auxiliary function (e.g., guessing whether the entities are the same or different).

In chapter 4, we introduce a new game mechanism called *complementary-agreement* (and a game called The Perfect Split) that allows us to extract attributes that can help discriminate between sets of entities (e.g., bird images belonging to different species). Given an attribute, this mechanism asks one player to select entities that have that attribute, while the other player is asked to do the opposite, i.e., select entities that do not have that attribute. The mechanism ensures output reliability by having each player generate outputs to constrain the other player's outputs. The complementary-agreement mechanism can be used to *discover* attributes: by presenting a set of entities to (one of) the players, and asking them to enter an attribute to split the images.

Both of these game mechanisms share a similar **query model** – instead of presenting one entity to the players, these games present a set of entities and ask players to exchange attributes that can discriminate between the entities.

2. We demonstrate the intricate dependency between the human-centric and task-centric objectives of human computation games. For example, while being human-friendly, the data collected by TagATune is extremely noisy. That is, the matrix generated by the crowd via TagATune ends up having a vast number of columns (i.e., attributes) that are redundant. This poses a problem for achieving the task-centric objective of the system, i.e., to complete the matrix using the help of machine learning algorithms, since most existing automated music tagging algorithms adopt the traditional approach of training multiple binary classifiers and are designed to handle a small number of mutually exclusive attributes as labels. In chapter 3, we propose an algorithm that can learn efficiently using this type of crowd-generated matrices for attribute learning. The algorithm is trained in two stages: first learn a topic model using the tags collected by TagATune for each music clip, then learn a mapping between audio features and the topic distribution assigned to each music clip by the topic model. This algorithm is both more (i) data-efficient (i.e., can utilize an arbitrarily large number of open vocabulary tags as training data) and time-efficient (i.e., reduces training time drastically), when compared to the traditional multiple binary classification approach.

3. We show a new use of games to elicit the help of humans to *evaluate* machine learning algorithms, i.e., having machines predict the values of the cells in the matrix, and ask humans to verify machine output by scoring the same cells. In chapter 3, we show that by having music tagging algorithms pose as game players in TagATune, we can evaluate their performance by observing the percentage of the times players are able to guess that the music clips given to them and their game partner (i.e., an algorithm) are the same. In chapter 4, we show that it is feasible to elicit the help of crowdworkers to supervise a

web mining system called NELL continuously.

4. Through experimentation, we show that human workers are poor at determining the values of attributes, when either the attributes or the entities themselves are unfamiliar. In chapter 4, we show that this knowledge limitation can be overcome by (i) changing the representation of the entity (e.g., by transforming noun phrases into images, in the case of noun phrase categorization), or (ii) asking humans to identify other more commonly known attributes and determine their values, using which to train machine learning algorithms to predict the values of the unfamiliar attributes (e.g., asking humans to provide perceptual attributes like "red head," and "long tail" and use them to predict if a bird image contains a "Finch" or a "Corvid").

5. in chapter 5, we describe the design space of human computation games for attribute learning, including the key components of such games and the design decisions associated with each of the components. We also place existing game mechanisms within the context of this design space, and offer a few suggestions on how to choose what game mechanism to apply when given a new attribute learning problem with certain characteristics.

6.1.2 Practical Contributions

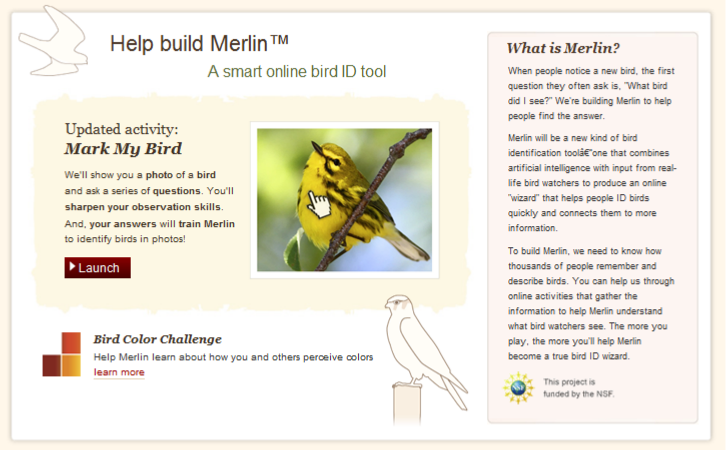
In addition to conceptual contributions, we implemented and deployed two actual systems that are shown to be effective at achieving both *human-centric* and *task-centric* objectives. In particular, TagATune has been deployed, interacted with tens of thousands of game players over several years, and generated a large amount of music tags that have been used by the research community to train automated music tagging algorithms. The complementary-agreement mechanism has been shown to be useful for encouraging truthful outputs on paid crowdsourcing platforms such as Mechanical Turk and in a game called The Perfect Split.

6.2 Future Directions

6.2.1 Interweaving Human and Machine Intelligence

The premise behind human computation is that given the right conditions, an unprecedented number of human workers can be mobilized to solve problems that are difficult for machines to solve. However, as a resource, human computers are still typically more costly than machines. Hybrid solutions, involving both human and machine intelligence, as we show in this thesis, can be powerful for future human computation systems.

Human computation games must select, for each round, entities to be presented to players. Some games, e.g., The Perfect Split, must additionally select the type of queries to ask using different modes. The games presented in this thesis do not make use of any intelligent policies for choosing entities or which modes of the game to



Help build Merlin™

A smart online bird ID tool

Updated activity:
Mark My Bird

We'll show you a **photo of a bird** and ask a series of **questions**. You'll **sharpen your observation skills**. And, **your answers** will **train Merlin** to identify birds in photos!

Bird Color Challenge

Help Merlin learn about how you and others perceive colors

What is Merlin?

When people notice a new bird, the first question they often ask is, "What bird did I see?" We're building Merlin to help people find the answer.

Merlin will be a new kind of bird identification toolâ€"one that combines artificial intelligence with input from real-life bird watchers to produce an online "wizard" that helps people ID birds quickly and connects them to more information.

To build Merlin, we need to know how thousands of people remember and describe birds. You can help us through online activities that gather the information to help Merlin understand what bird watchers see. The more you play, the more you'll help Merlin become a true bird ID wizard.

This project is funded by the NSF.

Figure 6.1: Mark My Bird

present next to players. In reality, information about different entities have different utilities, and different modes of the game can have differing cost (e.g., in terms of difficulty level for players) and benefit (e.g., in terms of asking for information that actually moves the system towards achieving its task-centric objective). An important next step in our work is to investigate some active learning or decision theoretic framework (similar to [Kamar 2012]) for automatically choosing entities and different modes to present to players during each round of the game.

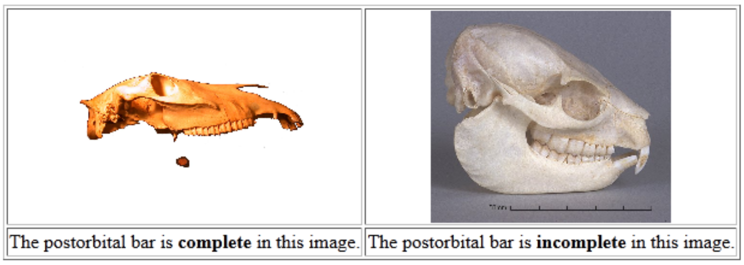
6.2.2 Applications in Citizen Science

Modern science is data intensive. In order to test hypotheses about our natural environment, e.g., about climate patterns, species distribution and trajectories of stars, often we need to collect and analyze data over large geographical regions and many time periods. When carried out by a few scientists, this process is tedious, time-consuming and sometimes impossible. The idea of citizen science is to engage non-scientists in the collection and interpretation of data in order to answer some scientific questions. Citizen science projects existed as early as 1900s where people volunteered to collect daily climate data at regular time intervals and report their measurements via telegraph [Grier 2005]. The National Audubon Society holds an annual Christmas Bird Count, which has lasted more than 100 years and engaged more than 50,000 birding enthusiasts in collecting bird count information in 2005 [Baron 2006]. The American Association of Variable Star Observers (AAVSO) [Williams 2001] has engaged amateur astronomers for more than a century in tracking the variation of brightness in stars.

With new Internet and mobile technologies at our fingertips, it is becoming easy for vast number of people all over the world to participate as citizen scientists, to collect field data to help answer specific scientific hypothesis, to map and monitor animal and plant species, and to annotate massive amount of scientifically interesting images and field recordings; all these are still beyond the capabilities of machines. Galaxy Zoo [Lintott 2008, Lintott 2010], for example, has more than 200,000 participants from 113 countries making more than 100 million classifications of galaxies [Raddick 2010], resulting in new discoveries [Cardamone 2009] and an expanded project called Zooinverse [Zooniverse 2012]. E-bird [EBird 2012], an Internet-based citizen science project run by Cornell Lab of Ornithology, has over a period of five years attracted over 500,000 users and collected 21 million bird records [Sullivan 2009].

Technologies can be used to bridge the knowledge gap of citizen scientists. For example, one of the biggest science-related human computation (i.e., citizen science) projects that exist is called e-Bird, launched in 2002 by the Cornell Lab of Ornithology at Cornell University and the National Audubon Society, where hundreds of thousands of birders submit observational data about birds they encountered and identified to a central database. Many birders are novices who might not be able to identify the correct species of the bird. A promising approach, as adopted by computer vision tools such as Visipedia [Perona 2010], is to allow users to upload the photo of a bird, which is then fed to a computer vision algorithm that determines its category. However, bird identification remains a challenging task for computer vision, and human computation becomes an attractive alternative.

For example, the *Merlin* project at Cornell's Lab of Ornithology is investigating an approach for bird identification, where birders can confirm the identity of a bird they saw in the field by entering visual attributes into a system, which then returns a set of candidate species and their images. In order for this identification tool to work, it is necessary to collect large amounts of attribute data about birds. Human computation games, such as Mark My Bird (Figure 6.1), are already being developed to collect color-related bird attributes for Merlin. In the future, we will investigate whether the data we collected via The Perfect Split can be used to power similar identification system for a variety of animal species.



The postorbital bar is **complete** in this image.

The postorbital bar is **incomplete** in this image.

Figure 6.2: Scoring unfamiliar attributes

Finally, the problem of learning unfamiliar attributes appear in many citizen science problems. Figure 6.2 shows an example of such task, where workers are given an unfamiliar attribute *postorbital bar* and asked to score the value of that attribute to be *complete* or *incomplete*. Here, there are added difficulties that the attribute is difficult to describe, and workers might require substantial training before they are capable of scoring the values of such an attribute accurately. In future work, we hope to investigate new methods for overcoming this type of knowledge limitation in attribute learning.

Bibliography

[Abbeel 2004] P. Abbeel and A.Y. Ng. *Apprenticeship learning via inverse reinforcement.* In ICML, 2004. (Cited on page 28.)

[Aucouturier 2006] J.J. Aucouturier. *Ten experiments on the modeling of Polyphonic Timbre.* PhD thesis, 2006. (Cited on page 56.)

[Banko 2007] M. Banko and O. Etzioni. *Strategies for lifelong knowledge extraction from the web.* In K-CAP, 2007. (Cited on pages 28 and 103.)

[Baron 2006] G. Baron. *The 106th christmas bird count.* `http://web4.audubon.org/bird/cbc/pdf/ab_106_001-09w%20jump1002final.pdf`, 2006. (Cited on page 128.)

[Bennett 2009] P. Bennett, M. Chickering and A. Mityagin. *Picture this: preferences for image search.* In Human Computation Workshop (HCOMP) 2009, 2009. (Cited on page 23.)

[Berg 2010] T. Berg, A. Berg and J. Shih. *Automatic attribute discovery and characterization from noisy web data.* In ECCV, 2010. (Cited on page 27.)

[Berger 1996] A. Berger, S. Della Pietra and V. Della Pietra. *A Maximum Entropy Approach to Natural Language Processing.* Computational Linguistics, vol. 22, no. 1, pages 39–71, 1996. (Cited on page 67.)

[Bergstra 2006a] J. Bergstra, A. Lacoste and D. Eck. *Predicting genre labels for artists using freedb.* In ISMIR, pages 85–88, 2006. (Cited on page 26.)

[Bergstra 2006b] J. Bergstra, A. Lacoste and D. Eck. *Predicting genre labels for artists using freedb.* In ISMIR, pages 85–88, 2006. (Cited on page 62.)

[Bertin-Mahieux 2008a] T. Bertin-Mahieux, D. Eck, F. Maillet and P. Lamere. *Autotagger: a model for predicting social tags from acoustic features on large music databases.* TASLP, vol. 37, no. 2, pages 115–135, 2008. (Cited on page 26.)

[Bertin-Mahieux 2008b] T. Bertin-Mahieux, D. Eck, F. Maillet and P. Lamere. *Autotagger: a model for predicting social tags from acoustic features on large music databases.* TASLP, vol. 37, no. 2, pages 115–135, 2008. (Cited on pages 33, 62 and 74.)

[Betteridge 2009] J. Betteridge, A. Carlson, S.A. Hong, E. Hruschka, E. Law, T. Mitchell and S.H. Wang. *Towards never ending language learning.* In AAAI Spring Symposium on Learning by Reading and Learning to Read, 2009. (Cited on page 28.)

[Blei 2003] D. Blei, A. Ng and M. Jordan. *Latent Dirichlet Allocation.* Journal of Machine Learning Research, vol. 3, pages 993–1022, 2003. (Cited on page 65.)

[Blei 2007] D. Blei and J.D. McAuliffe. *Supervised Topic Models.* In NIPS, 2007. (Cited on page 67.)

[Blei 2009] D. Blei and J. Lafferty. *Topic Models.* In A. Srivastava and M. Sahami, editeurs, Text Mining: Theory and Applications. Taylor and Francis, 2009. (Cited on page 67.)

[Blum 1998] A. Blum and T. Mitchell. *Combining labeled and unlabeled data with co-training.* In COLT, 1998. (Cited on page 28.)

[Branson 2010] S. Branson, C. Wah, B. Babenko, F. Schroff, P. Welinder, P. Perona and S. Belongie. *Visual recognition with humans in the loop.* In ECCV, 2010. (Cited on page 29.)

[Brodley 1999] C. Brodley and M. Friedl. *Identifying mislabeled training data.* JAIR, pages 131–167, 1999. (Cited on page 62.)

[Cardamone 2009] C. N. Cardamone, K. Schawinski, M. Sarzi, S. P. Bamford, N. Bennert, C. M. Urry, C. Lintott, W. C. Keel, J. Parejko, R. C. Nichol, D. Thomas, D. Andreescu, P. Murray, M. J. Raddick, A. Slosar, A. Szalay and J. VandenBerg. *Galaxy zoo green peas: Discovery of a class of compact extremely star-forming galaxies.* MNRAS, 2009. (Cited on page 129.)

[Carlson 2009] A. Carlson, J. Betteridge, E. Hruschka and T. Mitchell. *Coupling semi-supervised learning of categories and relations.* In NAACL-HLT Workshop on Semi-supervised Learning for Natural Language Processing, pages 1–9, 2009. (Cited on page 28.)

[Carlson 2010a] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. Hruschka and T. Mitchell. *Towards an architecture for neverending language learning.* In AAAI, 2010. (Cited on pages 28 and 103.)

[Carlson 2010b] A. Carlson, J. Betteridge, S. Wang, E. Hruschka and T. Mitchell. *Coupled semi-supervised learning for information extraction.* In WSDM, 2010. (Cited on pages 28, 40 and 103.)

[Chen 2009] L. Chen, P. Wright and W. Nejdl. *Improving music genre classification using collaborative tagging data.* In WSDM, pages 84–93, 2009. (Cited on page 33.)

[Csiszar 1996] I. Csiszar. *Maxent, mathematics, and information theory.* In K. Hanson and R. Silver, editeurs, Maximum Entropy and Bayesian Methods. Kluwer Academic Publishers, 1996. (Cited on page 67.)

[Culotta 2006] A. Culotta, T. Kristjansson, A. McCallum and P. Viola. *Corrective feedback and persistent learning for information extraction.* Fix, vol. 170, no. 14–15, pages 1101–1122, 2006. (Cited on page 28.)

[Dannenberg 2004] R.B. Dannenberg and N. Hu. *Understanding search performance in query-by-humming systems.* In ISMIR, pages 41–50, 2004. (Cited on page 33.)

[Doan 2011] A. Doan, R. Ramakrishnan and A. Y. Halevy. *Mass collaboration systems on the world-wide web.* Communications of the ACM, vol. 54, no. 4, 2011. (Cited on page 15.)

[Donmez 2008] P. Donmez and J. Carbonell. *Proactive learning: Cost-sensitive active learning with multiple imperfect oracles.* In CIKM, 2008. (Cited on page 30.)

[Donmez 2009a] P. Donmez and J. Carbonell. From active to proactive learning methods. springer: Studies in computational intelligence. 2009. (Cited on page 30.)

[Donmez 2009b] P. Donmez, J. Carbonell and J. Schneider. *Efficiently leanring the accuracy of labeling sources for selective sampling.* In KDD, 2009. (Cited on pages 30 and 31.)

[Donmez 2010a] P. Donmez, J. Carbonell and J. Schneider. *A probabilistic framework to learn from multiple annotators with time-varying accuracy.* In SDM, 2010. (Cited on page 31.)

[Donmez 2010b] P. Donmez, J. Carbonell and J. Schneider. *Probabilistic framework to learn from multiple annotators with time-varying accuracy.* In SIAM, 2010. (Cited on page 31.)

[Downie 2008] J.S. Downie. *The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research.* Acoustical Science and Technology, vol. 29, no. 4, pages 247–255, 2008. (Cited on pages 51 and 55.)

[EBird 2012] EBird. `http://ebird.org/`, 2012. (Cited on page 129.)

[Eisenberg 2004] G. Eisenberg, J.M. Batke and T. Sikora. *Beatbank – an mpeg-7 compliant query by tapping system.* In Audio Engineering Society Convention, page 6136, 2004. (Cited on page 33.)

[Erosheva 2004] E. Erosheva, S. Fienberg and J. Lafferty. *Mixed-membership models of scientific publications.* Proceedings of the National Academy of Sciences of the United States of America, vol. 101, no. Suppl 1, pages 5220–5227, April 2004. (Cited on page 65.)

[Farhadi 2009] A. Farhadi, I. Endres, D. Hoiem and D. Forsyth. *Describing objects by their attributes.* In CVPR, 2009. (Cited on page 27.)

[Farhadi 2010] A. Farhadi, I. Endres and D. Hoiem. *Attribute-centric recognition for cross-category generalization.* In CVPR, 2010. (Cited on page 27.)

[Fogarty 2008] J. Fogarty, D. Tan, A. Kapoor and S. Winder. *Cueflik: Interactive concept learning in image search.* In CHI, 2008. (Cited on page 29.)

[FriedmanTest 2012] FriedmanTest. *Friedman test.* http://en.wikipedia.org/wiki/Friedman_test, 2012. (Cited on page 99.)

[GoogleImageLabeler 2012] GoogleImageLabeler. http://images.google.com/imagelabeler/, 2012. (Cited on page 21.)

[Goto 2004] M. Goto and K. Hirata. *Recent Studies on Music Information Processing.* Acoustic Science and Technology, pages 419–425, 2004. (Cited on page 33.)

[Grauman 2011] K. Grauman and B. Leibe. *Visual Object Recognition.* Synthesis Lectures on Artificial Intelligence and Machine Learning, vol. 5, no. 2, pages 1–181, 2011. (Cited on page 27.)

[Grier 2005] D.A. Grier. When computers were human. Princeton University Press, 2005. (Cited on page 128.)

[Hacker 2009] S. Hacker and L. von Ahn. *Eliciting User Preferences with an Online Game.* In CHI, pages 1207–1216, 2009. (Cited on page 23.)

[Healey 2007] P. Healey, N. Swoboda, I. Umata and J. King. *Graphical language games: interactional constraints on representational form.* Cognitive Science, vol. 31, pages 285–309, 2007. (Cited on page 38.)

[Hearst 1999] M. Hearst, editeur. Mixed-initiative interaction: Trends and controversies, pages 14–23. 1999. (Cited on page 28.)

[Herrera 2003] P. Herrera, G. Peeters and S. Dubnov. *Automatic Classification of Music Instrument Sounds.* Journal of New Music Research, pages 3–21, 2003. (Cited on page 33.)

[Ho 2009] C.J. Ho, T.H. Chang, J.C. Lee, J. Hsu, and K.T. Chen. *Kisskissban: a competitive human computation game for image annotation.* In 1st Human Computation Workshop, pages 11–14, 2009. (Cited on page 91.)

[Hoffman 2009] M. Hoffman, D. Blei and P. Cook. *Easy as CBA: A simple probabilistic model for tagging music.* In ISMIR, pages 369–374, 2009. (Cited on pages 26, 33, 62, 73 and 74.)

[Hofmann 1999] T. Hofmann. *Probabilistic Latent Semantic Analysis.* In UAI, pages 50–57, 1999. (Cited on page 65.)

[Horton 2010] J. Horton, D. Rand and R. Zeckhauser. *The online laboratory.* In WINE, 2010. (Cited on page 19.)

[Horvitz 2007a] E. Horvitz. *Reflections on challenges and promises of mixed-initiative interaction.* AAAI Magazine, vol. 28, 2007. (Cited on page 29.)

[Horvitz 2007b] E. Horvitz and T. Paek. *Complementary computing: Policies for transferring callers from dialog systems to human receptionists.* User Modeling and User Adapted Interaction, vol. 17, 2007. (Cited on pages 16 and 29.)

[Horvitz 2008] E. Horvitz. *Artificial intelligence in the open world.* `http://research.microsoft.com/en-us/um/people/horvitz/aaai_presidential_lecture_eric_horvitz.htm`, 2008. (Cited on page 31.)

[Hu 2009] D. Hu and L. Saul. *A Probabilistic Topic Model for Unsupervised Learning of Musical Key-Profiles.* In ISMIR, pages 441–446, 2009. (Cited on page 65.)

[Hurwicz 1972] L. Hurwicz. On informationally decentralized systems. Decision and Organization: a Volume in Honor of Jacob Marshak. 1972. (Cited on page 22.)

[IGDAWhitePaper 2009] IGDAWhitePaper. `http://archives.igda.org/casual/IGDA_Casual_Games_White_Paper_2008.pdf`, 2009. (Cited on pages 20 and 21.)

[Ipeirotis 2010a] P. Ipeirotis. *Analyzing the amazon mechanical turk marketplace.* XRDS, vol. 17, no. 2, 2010. (Cited on page 18.)

[Ipeirotis 2010b] P. Ipeirotis. *Demographics of mechanical turk.* Rapport technique, New York University, 2010. (Cited on page 19.)

[Iwata 2009] T. Iwata, T. Yamada and N. Ueda. *Modeling Social Annotation Data with Content Relevance using a Topic Model.* In NIPS, 2009. (Cited on page 63.)

[Jackson 2003] M. O. Jackson. Mechanism theory. Encyclopedia of Life Support Systems. EOLSS Publishers, 2003. (Cited on page 22.)

[Jain 2008] S. Jain and D. Parkes. *A game-theoretic analysis of games with a purpose.* In WINE, pages 342–350, 2008. (Cited on page 34.)

[Joachims 2005] T. Joachims, L. Granka, B. Pan, H. Hembrooke and G. Gay. *Accurately interpreting clickthrough data as implicit feedback.* In SIGIR, 2005. (Cited on page 29.)

[Kaelbling] L. Kaelbling, M. Littman and A. Moore. *Reinforcement learning: A survey.* Fix, vol. 4, pages 237–285. (Cited on page 30.)

[Kaelbling 1993] L. Kaelbling. Learning in embedded systems. The MIT Press, 1993. (Cited on page 30.)

[Kamar 2012] E. Kamar, S. Hacker and E. Horvitz. *Combining Human and Machine Intelligence in Large-scale Crowdsourcing.* In AAMAS, 2012. (Cited on pages 29 and 128.)

[Kapoor 2007] A. Kapoor and E. Horvitz. *Principles of lifelong learning for predictive user modeling.* In UM, 2007. (Cited on page 29.)

[Kapoor 2008] A. Kapoor and E. Horvitz. *Experience sampling for building predictive user models: a comparative study.* In ICML, 2008. (Cited on page 28.)

[Kapoor 2012] A. Kapoor, B. Lee, D. Tan and E. Horvitz. *Performance and Preferences: Interactive Refinement of Machine Learning Procedures.* In AAAI, 2012. (Cited on page 29.)

[Kapur 2004] A. Kapur, M. Benning and G. Tzanetakis. *Query by Beatboxing.* In ISMIR, pages 170–178, 2004. (Cited on page 33.)

[Kearn 2006] M. Kearn, S. Suri and N. Montfort. *An experimental study of the coloring problem on human subject networks.* Science, vol. 313, no. 5788, pages 824–827, August 2006. (Cited on page 14.)

[Kemp 2006] C. Kemp, J.B. Tenenbaum, T.L. Griffiths, T. Yamada and N. Ueda. *Learning systems of concepts with an infinite relational model.* In AAAI, 2006. (Cited on page 27.)

[Kim 2008] Y.E. Kim, E. Schmidt and L. Emelle. *Moodswings: A collaborative game for music mood label collection.* In ISMIR, pages 231–236, 2008. (Cited on pages 5, 34 and 44.)

[Kittur 2007] A. Kittur, B. Suh, B. Pendleton and E. Chi. *He says, she says: Conflict and coordination in wikipedia.* In CHI, 2007. (Cited on page 15.)

[Kittur 2008] A. Kittur and R. Kraut. *Harnessing the wisdom of crowds in wikipedia: Quality through coordination.* In CHI, 2008. (Cited on page 15.)

[Kramer 1956] C.Y. Kramer. *Extension of multiple range tests to group means with unequal number of replications.* Biometrics, vol. 12, pages 307–310, 1956. (Cited on page 56.)

[Kumar 2009] N. Kumar, A.C. Berg, P.N. Belhumeur and S.K. Nayar. *Attribute and simile classifiers for face verification.* In ICCV, 2009. (Cited on page 27.)

[Lamere 2008a] P. Lamere. *Social tagging and music information retrieval.* Journal of New Music Research, vol. 37, no. 2, pages 101–114, 2008. (Cited on pages 27 and 33.)

[Lamere 2008b] P. Lamere. *Social Tagging and Music Information Retrieval.* Journal of New Music Research, vol. 37, no. 2, pages 101–114, 2008. (Cited on page 63.)

[Lampert 2009] C.H. Lampert, H. Nickisch and S. Harmeling. *Learning to detect unseen object classes by between-class attribute transfer.* In CVPR, 2009. (Cited on page 27.)

[Laurier 2009] C. Laurier, M. Sordo, J. Serra and P. Herrera. *Music Mood Representations from Social Tags.* In ISMIR, pages 381–386, 2009. (Cited on page 63.)

[Law 2007] E. Law, L. von Ahn, R. Dannenberg and M. Crawford. *Tagatune: A game for music and sound annotation.* In ISMIR, 2007. (Cited on page 35.)

[Law 2008] E. Law. *The Problem of Accuracy as an Evaluation Criterion.* In Proceedings of the ICML Workshop on Evaluation Methods in Machine Learning, pages 1–4, 2008. (Cited on page 50.)

[Law 2009a] E. Law, A. Mityagin and M. Chickering. *Intentions: A game for classifying search query intent.* In CHI Work in Progress, 2009. (Cited on page 50.)

[Law 2009b] E. Law and L. von Ahn. *Input-agreement: A new mechanism for collecting data using human computation games.* In CHI, pages 1197–1206, 2009. (Cited on pages 61, 63 and 69.)

[Law 2011a] E. Law and L. von Ahn. Human computation: Synthesis lectures on artificial intelligence and machine learning. Morgan and Claypool, 2011. (Cited on page 13.)

[Law 2011b] E. Law and H. Zhang. *Towards large-scale collaborative planning: Answering high-level search queries using human computation.* In AAAI, 2011. (Cited on page 29.)

[Law 2012] E. Law. *Magnatagatune Dataset.* `http://tagatune.org/Magnatagatune`, 2012. (Cited on page 44.)

[Lee 2009] B. Lee and L. von Ahn. *Squigl.* `http://www.gwap.com/gwap/gamesPreview/squigl/`, 2009. (Cited on page 23.)

[Levy 2007] M. Levy and M. Sandler. *A Semantic Space for Music Derived from Social Tags.* In ISMIR, 2007. (Cited on page 63.)

[Lewis 1994] D. Lewis and J. Catlett. *Heterogeneous uncertainty sampling for supervised learning.* In ICML, pages 148–156, 1994. (Cited on pages 28 and 30.)

[Li 2003] T. Li, M. Ogihara and Q. Li. *A comparative study on content-based music genre classification.* In SIGIR, pages 282–289, 2003. (Cited on pages 26 and 62.)

[Lintott 2008] C. J. Lintott, K. Schawinski, A. Slosar, K. Land, S. Bamford, D. Thomas, M.J. Raddick, R. Nichol, A. Szalay, D. Andreescu, P. Murray and J. van den Berg. *Galaxy Zoo: Morphologies derived from visual* *inspection of galaxies from the Sloan Digital Sky Survey.* Monthly Notices of the Royal Astronomical Society (MNRAS), vol. 389, no. 3, pages 1179–1189, April 2008. http://arxiv.org/abs/0804.4483. (Cited on page 129.)

[Lintott 2010] C. Lintott, K. Schawinski, S. Bamford, A. Slosar, K. Land, D. Thomas, E. Edmondson, K. Masters, R. Nichol, J. Raddick, A. Szalay, D. Andreescu, P. Murray and J. Vandenberg. *Galaxy zoo 1 : Data release of morphological classifications for nearly 900,000 galaxies.* MNRAS, 2010. (Cited on page 129.)

[Ma 2009] H. Ma, R. Chandrasekar, C. Quirk and A. Gupta. *Improving Search Engines Using Human Computation Games.* In CIKM, 2009. (Cited on page 50.)

[Malone 1998] T.W. Malone and R.J. Laubacher. *The dawn of the e-lance economy.* arvard Business Review, vol. 76, no. 5, pages 144–152, 1998. (Cited on page 15.)

[Mandel 2005] M. Mandel and D. Ellis. *Song-level features and support vector machines for music classification.* In ISMIR, 2005. (Cited on pages 26 and 62.)

[Mandel 2009a] M. Mandel and D. Ellis. *Labrosa's Audio Classification Submissions.* www.music-ir.org/mirex/2008/abs/AA_AG_AT_MM_CC_mandel.pdf, 2009. (Cited on page 70.)

[Mandel 2009b] M. Mandel and D. Ellis. *A web-based game for collecting music metadata.* Journal of New Music Research, vol. 37, no. 2, pages 151–165, 2009. (Cited on pages 5, 23, 31, 34, 44 and 50.)

[Mason 2010] W. Mason and S. Suri. *Conducting behavioral research on amazon's mechanical turk.* Rapport technique, Social Science Research Network Working Paper Series, 2010. (Cited on page 19.)

[Maytal 2009] S. Maytal, P. Melville and F. Provost. *Active feature-value acquisition.* Management Science, vol. 55, no. 4, pages 664–684, 2009. (Cited on page 28.)

[McCallum 2002] A. McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002. (Cited on page 67.)

[Mimno 2008] D. Mimno and A. McCallum. *Topic Models Conditioned on Arbitrary Features with Dirichlet-multinomial Regression.* In UAI, 2008. (Cited on page 67.)

[Mirex 2012] Mirex. http://www.music-ir.org/mirex/wiki/MIREX_HOME, 2012. (Cited on page 51.)

[Mohamed 2011] T.P. Mohamed, E. Hruschka and T.M. Mitchell. *Discovering relations between noun categories.* In EMNLP, 2011. (Cited on page 103.)

[MTurk 2012] MTurk. https://www.mturk.com/mturk/welcome, 2012. (Cited on page 18.)

[MTurkRequestorInterface 2012] MTurkRequestorInterface. https://requester.mturk.com, 2012. (Cited on page 19.)

[Nigam 1999] K. Nigam, J. Lafferty and A. Mccallum. *Using Maximum Entropy for Text Classification.* In IJCAI-99 Workshop on Machine Learning for Information Filtering, pages 61–67, 1999. (Cited on page 67.)

[Nisan 2007] N. Nisan. Introduction to mechanism design (for computer scientists). Algorithmic Game Theory. Cambridge University Press, 2007. (Cited on page 22.)

[Nocedal 1995] J. Nocedal. *Updating quasi-Newton matrices with limited storage.* Mathematics of Computation, pages 339–353, 1995. (Cited on page 67.)

[North 1990] D. North. *Readings in uncertain reasoning.* chapitre A tutorial introduction to decision theory, pages 68–78. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990. (Cited on page 30.)

[O. Chapelle 2006] B. Scholkopf O. Chapelle and A. Zien, editeurs. Semi-supervised learning. MIT Press, 2006. (Cited on page 28.)

[Palatucci 2009] M. Palatucci, D. Pomerleau, G. Hinton and T. Mitchell. *Zero-shot learning with semantic output codes.* In NIPS, 2009. (Cited on pages 27, 64 and 65.)

[Paolacci 2010] G. Paolacci, J. Chandler, and P. Ipeirotis. *Running experiments on amazon mechanical turk.* Judgment and Decision, vol. 5, no. 5, pages 411–419, 2010. (Cited on page 19.)

[Parikh 2011] D. Parikh. *Interactively building a discriminative vocabulary of nameable attributes.* In CVPR, 2011. (Cited on page 27.)

[Pereira 2006] F. Pereira and G. Gordon. *The Support Vector Decomposition Machine.* In Proceedings of the 23^{rd} International Conference on Machine Learning (ICML), pages 689–696, 2006. http://portal.acm.org/citation.cfm?id=1143844.1143931. (Cited on page 67.)

[Perona 2010] P. Perona. *Vision of a Visipedia.* IEEE, vol. 98, no. 8, pages 1526–1534, 2010. (Cited on page 129.)

[Raddick 2010] M.J. Raddick, G. Bracey, P.L. Gay, C.J. Lintott, P. Murray, K. Schawinski, A.S. Szalay and J. Vandenberg. *Galaxy zoo: Exploring the motivations of citizen science volunteers.* MNRAS, 2010. (Cited on page 129.)

[ReadTheWeb 2012] ReadTheWeb. http://rtw.ml.cmu.edu/wsdm10_online/, 2012. (Cited on page 28.)

[Rebbapragada 2007] U. Rebbapragada and C. Brodley. *Class Noise Mitigation Through Instance Weighting.* In ECML, pages 708–715, 2007. (Cited on page 62.)

[Rohrbach 2010] M. Rohrbach, M. Stark, G. Szarvas, I. Gurevych and B. Schiele. *What helps where – and why? semantic relatedness for knowledge transfer.* In CVPR, 2010. (Cited on page 27.)

[Rosenthal 2010] S. Rosenthal. *Human Modeling and Interaction for Effective Task Autonomy.* PhD thesis, 2010. (Cited on pages 28 and 29.)

[Ross 2010] J. Ross, L. Irani, M.S. Silberman, A. Zaldivar and B. Tomlinson. *Who are the crowdworkers? shifting demographics in mechanical turk.* In Work-in-Progress, pages 2863–2872, 2010. (Cited on pages 18, 19 and 20.)

[Roy 2001] N. Roy and A. McCallum. *Toward optimal active learning through sampling estimation of error reduction.* In ICML, pages 441–448, 2001. (Cited on page 30.)

[Seemakurty 2010] N. Seemakurty, J. Chu, L. von Ahn and A. Tomasic. *Word sense disambiguation via human computation.* In 2nd KDD Human Computation Workshop, 2010. (Cited on page 23.)

[Settles 2012] B. Settles. *Active Learning.* Synthesis Lectures on Artificial Intelligence and Machine Learning, vol. 6, no. 1, pages 1–114, 2012. (Cited on pages 15, 28 and 29.)

[Seung 1992] H. Seung, M. Opper and H. Sompolinsky. *Query by committee.* In ACM Workshop on Computational Learning Theory, pages 287–294, 1992. (Cited on page 30.)

[Shahaf 2010] D. Shahaf and E. Horvitz. *Generalized task markets for human and machine computation.* In AAAI, 2010. (Cited on page 29.)

[Shilman 2006] M. Shilman, D. Tan and P. Simard. *CueTIP: A Mixed-Initiative Interface for Correcting Handwriting Errors.* In UIST, pages 323–332, 2006. (Cited on pages 28 and 29.)

[Singh 2008] A. Singh and G. Gordon. *Relational learning via collective matrix factorization.* In KDD, pages 650–658, 2008. (Cited on page 67.)

[Siorpaes 2008] K. Siorpaes and M. Hepp. *Games with a purpose for the semantic web.* IEEE Intelligent Systems, vol. 23, no. 3, pages 50–60, 2008. (Cited on page 23.)

[Speer 2010] R. Speer, C. Havasi and H. Surana. *Using verbosity: Common sense data from games with a purpose.* In FLAIRS, 2010. (Cited on pages 24 and 25.)

[Steyvers 2007] M. Steyvers and T. Griffiths. *Probabilistic Topic Models.* In T. Landauer, D.S. McNamara, S. Dennis and W. Kintsch, editeurs, Handbook of Latent Semantic Analysis. Erlbaum, Hillsdale, NJ, 2007. (Cited on pages 65 and 67.)

[Sullivan 2009] B.L. Sullivan, C.L. Wood, M.J. Iliff, R.E. Bonney, D. Fink and S. Kelling. *ebird: A citizen-based bird observation network in the biological sciences.* Biological Conservation, 2009. (Cited on page 129.)

[Tamuz 2011] O. Tamuz, C. Liu, S. Belongie, O. Shamir and A. Kalai. *Adaptively Learning the Crowd Kernel.* In ICML, 2011. (Cited on page 29.)

[Thaler 2011] S. Thaler, E. Simperl, K. Siorpaes and C. Hofer. *A survey on games for knowledge acquisition.* Rapport technique, STI, 2011. (Cited on page 26.)

[Thomas 2006] A.L. Thomas and C. Breazeal. *Reinforcement learning with human teachers: understanding how people want to teach robots.* In AAAI, 2006. (Cited on page 28.)

[Trohidis 2008] K. Trohidis, G. Tsoumakas, G. Kalliris and I. Vlahavas. *Multi-label classification of music emotions.* In ISMIR, pages 325–330, 2008. (Cited on page 33.)

[Tukey 1953] J.W. Tukey. *The problem of multiple comparisons.* Princeton University, 1953. (Cited on page 56.)

[Turing 1950] A. M. Turing. *Computing machinery and intelligence.* Mind LIX, vol. 2236, pages 43–60, October 1950. (Cited on page 13.)

[Turnbull 2007a] D. Turnbull, L. Barrington, D. Torres and G. Lanckriet. *Towards music query-by-semantic description using the CAL500 data set.* In SIGIR, pages 439–446, 2007. (Cited on pages 26 and 62.)

[Turnbull 2007b] D. Turnbull, R. Liu, L. Barrington and G. Lanckriet. *A game-based approach for collecting semantic annotations for music.* In ISMIR, pages 535–538, 2007. (Cited on pages 23, 31, 34, 44 and 50.)

[Turnbull 2007c] D. Turnbull, R. Liu, L. Barrington and G. Lanckriet. *A game-based approach for collecting semantic annotations of music.* In ISMIR, pages 535–538, 2007. (Cited on page 61.)

[Turnbull 2008a] D. Turnbull, L. Barrington, D. Torres and G. Lanckriet. *Five approaches to collecting tags for music.* In ISMIR, 2008. (Cited on page 27.)

[Turnbull 2008b] D. Turnbull, L. Barrington, D. Torres and G. Lanckriet. *Semantic annotation and retrieval of music and sound effects.* TASLP, vol. 16, no. 2, pages 467–476, 2008. (Cited on pages 26, 33, 62, 64 and 74.)

[Tzanetakis 2001] G. Tzanetakis, G. Essl and P. Cook. *Automatic music genre classification of audio signals.* In ISMIR, 2001. (Cited on page 33.)

[Tzanetakis 2002] G. Tzanetakis and P. Cook. *Musical genre classification of audio signals.* IEEE Transactions on Speech and Audio Processing, vol. 10, no. 5, pages 293–302, 2002. (Cited on page 33.)

[Vickrey 2008] D. Vickrey, A. Bronzan, W. Choi, A Kumar, J. Turner-Maier, A. Wang, and D. Koller. *Online word games for semantic data collection.* In EMNLP, 2008. (Cited on pages 23 and 34.)

[von Ahn 2004] L. von Ahn and L. Dabbish. *Labeling images with a computer game.* In CHI, pages 319–326, 2004. (Cited on pages 21, 22, 34 and 35.)

[von Ahn 2006a] L. von Ahn, R. Liu, and M. Blum. *Peekaboom: A game for locating objects in images.* In CHI Notes, pages 55–64, 2006. (Cited on page 24.)

[von Ahn 2006b] L. von Ahn, R. Liu and M. Blum. *Verbosity: A game for collecting common-sense knowledge.* In CHI Notes, pages 75–78, 2006. (Cited on page 24.)

[von Ahn 2008a] L. von Ahn. *Human Computation.* PhD thesis, Carngie Mellon University, 2008. (Cited on page 13.)

[von Ahn 2008b] L. von Ahn and L. Dabbish. *Designing games with a purpose.* Communications of the ACM, vol. 51, no. 8, pages 58–67, 2008. (Cited on pages 17, 25 and 37.)

[Weber 2009] I. Weber, S. Robertson and M. Vojnovic. *Rethinking the esp game.* In CHI, pages 3937–3942, 2009. (Cited on page 34.)

[Welinder 2010] P. Welinder, S. Branson, S. Belongie and P. Perona. *The multidimensional wisdom of crowds.* In NIPS, pages 1–9, 2010. (Cited on page 31.)

[Whitman 2002] B. Whitman and P. Smaragdis. *Combining musical and cultural features for intelligent style detection.* In ISMIR, 2002. (Cited on page 33.)

[Wikipedia 2012] Wikipedia. `http://wikipedia.com`, 2012. (Cited on pages 5 and 14.)

[WikipediaMAPDefinition 2012] WikipediaMAPDefinition. `http://en.wikipedia.org/wiki/Information_retrieval`, 2012. (Cited on page 80.)

[Williams 2001] T.R. Williams. *Reconsidering the history of the aavso.* Journal of the American Association of Variable Star Observers, vol. 29, page 132, 2001. (Cited on page 128.)

[Wu 2010] F. Wu and D.S. Weld. *Open information extraction using Wikipedia.* In ACL, 2010. (Cited on page 28.)

[Yao 2009] L. Yao, D. Mimno and A. McCallum. *Efficient Methods for Topic Model Inference on Streaming Document Collections*. In KDD, pages 937–946, 2009. (Cited on pages 67 and 72.)

[Yue 2009] Y. Yue and T. Joachims. *Interactively optimizing information retrieval systems as a dueling bandits problem*. In ECML, 2009. (Cited on page 29.)

[Zhu 2003] X. Zhu, X. Wu and S. Chen. *Eliminating class noise in large datasets*. In ICML, pages 920–927, 2003. (Cited on page 62.)

[Zooniverse 2012] Zooniverse. `http://www.zooniverse.org`, 2012. (Cited on page 129.)

ML

MACHINE LEARNING
DEPARTMENT

Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213

Carnegie Mellon.