Attribute Learning Using Joint Human and Machine Computation

Edith Law

April 2011

Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213

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

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Abstract

Human computation is the study of systems where humans perform a major part of the computation or are an integral part of the overall computational process. 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 image descriptors, which can then used to index images and facilitate search. However, most existing human computation systems operate without any machine intervention. Likewise, very few supervised learning systems are taking advantage of these powerful new platforms to elicit help from human teachers. It is therefore largely unknown what *more* a human computation system can achieve with machines in the loop.

This thesis is centered around the problem of attribute learning – using the joint effort of human game players and machine learning algorithms 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 particular, our work focuses on two aspects of the problem – *how* to acquire attributes and attribute values from human computers using incentive-compatible game mechanisms, and *what* active learning strategies to employ for attribute and attribute value acquisition.

In addition to exploring these research issues, this thesis is expected to make practical contributions by creating several systems and datasets for attribute learning. These include (i) the TagATune and Polarity game, (ii) large datasets of objects (e.g., music, images, named entities) and their attributes learned by our system, (iii) a prototype information retrieval system called ELF (Entity Lookup-Finder) that allows users to search for objects by issuing complex queries in the form of concatenated attributes, or by answering a set of system-issued questions about the visual and non-visual attributes of the objects they are seeking.

Contents

1 Introduction

Human computation is the study of systems where humans perform a major part of the computation or are an integral part of the overall computational process. 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 image descriptors, which was then used to index images and facilitate search. However, most existing human computation systems operate without any machine intervention. Likewise, very few supervised learning systems are taking advantage of these powerful new platforms to elicit help from human teachers. It is therefore largely unknown what *more* a human computation system can achieve with machines in the loop.

This thesis is centered around the problem of attribute learning – using the joint effort of human game players and machine learning algorithms 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". There are three central aspects to the problem. The *how* aspect concerns the invention of new incentive-compatible game mechanisms for extracting attributes and attribute values. In particular, we introduce a new class of mechanisms called set-based function computation games, where players are presented with a set of objects, and asked to generate attributes that distinguish between the objects. Specifically, we focus on two set-based function computation games – Tagatune (which implements the input-agreement mechanism) and Polarity (which implements the complementary-agreement mechanism) – and study their inherent properties. The *what* aspect looks beyond a purely "human" computation solution, by employing appropriate active learning strategies to intelligently select the best set of objects to present to players in order to achieve some pre-defined computational objectives. Finally, the *who* aspect pertains to task routing –the question of how the system can take into account the objectives and characteristics of the game players, by assigning them tasks that are appropriate for their levels of competence and expertise. In this thesis, we will focus on the **how** and **what** aspects, leaving the who aspect as optional extension.

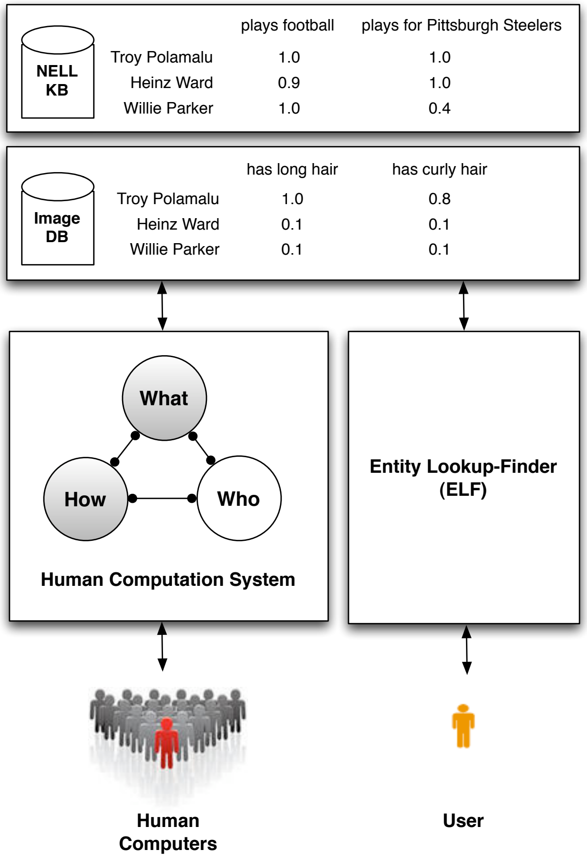
In addition to exploring these research issues, this thesis is expected to make several practical contributions by creating several systems and datasets for attribute learning. These include (i) the TagATune and Polarity game, (ii) large datasets of objects (e.g., music, images, named entities) and their attributes learned by our system, (iii) a prototype information retrieval system called ELF (which stands for Entity Lookup-Finder), which allows users to search for objects by issuing complex queries in the form of concatenated attributes, or by answering a set of system-issued questions about the visual and non-visual attributes of the objects they are seeking.

1.1 Motivation

The goal of this thesis is to work towards a fully integrated, machine-in-the-loop human computation system that can intelligently and continuously learn object attributes. Figure 1 illustrates such a system, consisted of (1) databases of objects (e.g., images and named entities) and the current beliefs about their attributes, (2) a human computation system consisted of games (*how*), an active learner for choosing what input objects to assign to game players (*who*), and a task router which takes into account players' competence and expertise (*who*), and (3) an information retrieval system which showcases how the attributes learned by our hybrid system can be used to support various retrieval and identification tasks. Consider the following scenarios as motivation for why such an integrated system will 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 called ELF, where Alice can 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 in ELF, e.g., "what movie did she star in?", "does she have brown hair?", "does she have thick eyebrows?" Bob is able to retrieve a set of candidate actresses along their headshots, and find the name of the actress he is looking for.



NELL KB	plays football	plays for Pittsburgh Steelers
Troy Polamalu	1.0	1.0
Heinz Ward	0.9	1.0
Willie Parker	1.0	0.4

Image DB	has long hair	has curly hair
Troy Polamalu	1.0	0.8
Heinz Ward	0.1	0.1
Willie Parker	0.1	0.1

Figure 1: The Vision

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". Using ELF, Eve is able to issue a complex search query and find a set of birds that fit those descriptions.

Scenario 2: Locating Objects
Co-bot is a robot that helps people, e.g., fetching objects that a person has misplaced. Imagine the following interaction, where Tom is looking for his cup and must describe to Co-bot what he is looking for. For example, Co-bot might ask "is a coffee mug?", "is it dark in color", "does it have writings on it?". Equipped with those answers, Co-bot can go around the room to try to classify each object by those attributes, and locate Tom's cup.

1.2 Hypothesis

The central thesis of this work is the following:

> *We can accurately and efficiently learn object attributes using the joint efforts of human computers and machine learning algorithms, by employing appropriate incentive-compatible game mechanisms (how) and active learning strategies for selecting input objects to present to human computers (what).*

1.3 Approach

To support this thesis, we will test multiple hypotheses related to the *how*, *what* and *who* (optional) aspects of the problem in an incremental manner. Taking the knowledge we gained from working with TagATune (Hypothesis How.1a, How.1b, How.1c, How.1d), we will study the Polarity game mechanism and its effectiveness at extracting and evaluating visual attributes (Hypothesis How.2a, How.2b, How.2c, How.2d) and non-visual attributes (Hypothesis How.3a, How.3b, How.3c, How.3d). Having demonstrated the utility of Polarity, we will then investigate more intelligent ways of choosing the set of objects in each round based on some other learning or computational objectives (Hypothesis What.1, What.2). Finally, we will describe the **optional** work on task routing, which involves a few smaller studies to understand task difficulty and the role of expertise in human computation (Hypothesis Who.1, Who.2). We expect that answering these hypotheses will yield important insights about the challenges faced by most hybrid human computation and machine learning systems.

1.4 Expected Contribution

This thesis is expected to help advance the research in attribute learning in multiple domains, by creating new interfaces, algorithms, and datasets. In addition, our work is a case study of building large-scale human computation systems with machine intelligence in the loop, and the challenges associated with such an endeavor. The theoretical and practical contributions of this work include

- new game mechanisms for extracting attributes and attribute values,
- new active learning algorithms for attribute acquisition,
- new understanding about three central aspects of machine-in-the-loop human computation systems,
- new datasets containing objects and their attributes, which will be released to various research communities.

2 Background

2.1 Attribute Learning

The word *attribute* (e.g., has wings) prefers to the intrinsic property of a concept (e.g., bird) [63]. Attributes can also be viewed as an *unary relation*, a binary relation (e.g., X plays for the Y) where one of arguments is fixed. 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 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).

Due to the proliferation of images, music and articles on the Web, there has been automated techniques for extracting and learning attributes. In this section, we will review the related work in this area.

2.1.1 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 [51, 52], Gaussian Mixture Models [76, 78], Boosting [10], Logistic Regression [9], and other probabilistic models [29]. 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 [43]. 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 [77].

2.1.2 Images

In Computer Vision, 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 [61] and visual object recognition [24, 25, 42, 44]. The second benefit is explanatory power – the learning system can now describe to users the reasons behind its predictions.

With the exception of [62], 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. [42] 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 [40, 60]. The outdoor scene datasets provided by [24, 25] 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 [8, 66] without any human supervision.

2.1.3 Named Entities

NELL (which stands for Never-Ending Language Learner) [11, 13, 15] 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 [12, 15, 16] 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 ([15, 64]) has already learned to extract a knowledge base containing tens of thousands of assertions with an accuracy around 85%. 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" [14] than the day before. In the context of NELL, we use the word *attributes* refer to either cateogories (e.g., "dog", "athlete") or relations (e.g., "has tail", "plays football").

The 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 can suffer from the accumulation of errors if left unsupervised [14]. For example, the class "baked goods" suffers from a concept drift – due to the incorrectly promoted named entity "internet cookie", the class "baked goods" became one that is related to computers instead of pastries. In addition, the NELL ontology 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 [15] 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 [57]. An important next step for NELL, therefore, is to consider what new thing to learn next [7]. These observations all point to the need for human supervision in NELL. One goal of this thesis is to evaluate the extent to which a game like Polarity can provide feedback that can help mitigate compounded errors and allow for continuous learning of concepts in the open world [31].

2.2 Human Computation

Since the use of the term in 2006, the phrase "Human Computation" has become synonymous with many other equally loosely defined research areas, such as "crowdsourcing", "social computing", "socio-computational systems", "collective intelligence". To understand what we mean by "Human Computation," we must first define for ourselves the word "computation." In our formulation, **computation** is the process of mapping of some input representation to some output representation using a explicit, finite set of instructions (i.e., an *algorithm*). Following this definition, "human computation" is simply computation that is carried out by a human. Likewise, "human computation" systems can be defined as intelligent systems that organizes humans to carry out the process of computation – whether it be performing the basic operations (or units of computation), taking charge of the control process itself (e.g., decide what operations to execute next or when to halt the program), or even synthesizing the program itself (e.g., by creating new operations and specifying how they are ordered). The objective of a human computation system is to find an accurate solution for a pre-specified computational problem in the most efficient way. An important element in our definitions is the idea of explicit control - that unlike other crowd-driven systems (e.g., Wikipedia), computation is not the consequence of the natural dynamics in a crowd, but the consequence of a deliberate algorithm.

Figure 2 outlines three central aspects of a human computation system where explicit control can be applied – including *what* (the algorithm that specifies what operations to perform and in what order), *how* (interfaces and mechanisms for querying human computers) and *who* (the task router that decides to whom to assign each computational task).

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 (e.g., repetitions, iterations, conditions) that specify how the operations are to be arranged and executed. For example, the ESP Game implements an algorithm that runs a set of atomic image-to-tag operations to label images. Similar

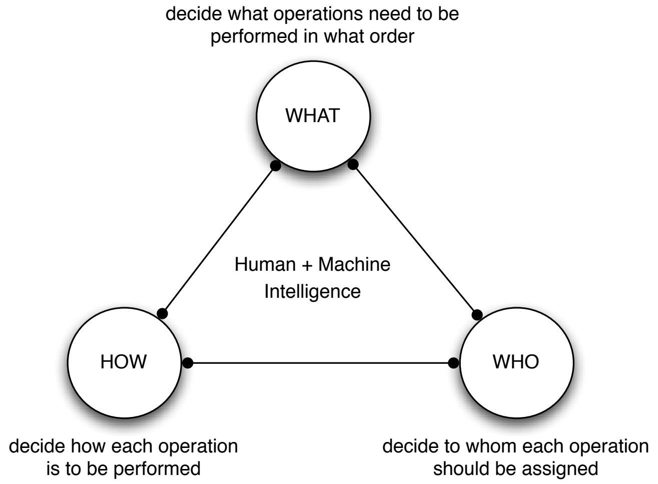

decide what operations need to be performed in what order
WHAT
Human + Machine Intelligence
HOW
WHO
decide how each operation is to be performed
decide to whom each operation should be assigned

Figure 2: Three Aspects of Machine-in-the-loop Human Computation Systems

to algorithms in the traditional sense, some human computation algorithms are more efficient than others. If our computational problem is to map a set of images to tags, a more efficient algorithm would make use of machine intelligence (e.g., active learning [70]) to select only images that the computer vision algorithm does not already know how to classify. Such an algorithm would 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 includes:

- 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 [30] to make computation more accurate and efficient?
- How do we decompose complex tasks into operations and order them in such a way to handle the idiosyncrasy of human workers?

Knowing what operations need to be performed, the next question is to whom. 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 to be a better (and faster) judge than someone without any medical training. Some research questions relevant to the **who** aspect of human computation includes:

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

Finally, the **how** aspect pertains to the question of design - how can the system that workers interact with motivates them to participate and to carry out the computational tasks to their best abilities (i.e., truthfully, accurately, and efficiently). Some research questions relevant to the "how" aspect of human computation includes:

- How do we design game mechanisms [82] that incentivize workers to tell the truth, i.e., generate accurate outputs?
- 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 community)?
- What are some new markets, organizational structures or interaction models for defining how workers relate to each other (as opposed to working completely independently)?

2.2.1 Games with a Purpose

As a human computation system, the ESP Game is 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. Since then, this data collection mechanism has been adopted for other games in the domain of image tagging (e.g. Matchin [27], Squigl [49], PictureThis [56]), music tagging (e.g. The Listen Game [18], MajorMiner [53], MoodSwings [41]) and semantic web ontologization (e.g. Ontogame, OntoTube [73]). This rapid adoption is evidence that a human computation game can be described by its general mechanism, which can then be readily extended to handle all kinds of data. The ESP game mechanism, for example, is referred to as output-agreement [81] because players are given the same input and must agree on an appropriate output. All games described above use the same mechanism as the ESP Game: match on the output.

The term *mechanism* [33] is borrowed from a field of research called *mechanism design*, 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 set of rules in which each participant benefits the most by sharing 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. Since we do not have ground truth, a mechanism usually involves *multiple* human computers and rules about how their outputs will jointly determine an outcome (i.e., reward or penalty). Besides the fact games have the ability to easily attract hundreds and thousands of human computers, the most important take-home message of the ESP game is this: *by setting up an incentive compatible mechanism involving multiple game players, we can ensure that the outputs they generate are accurate*.

Mechanisms, however, are not one-size-fits-all. Depending on the type of input data, output-agreement is neither the only nor the best mechanism for data collection. It has been noted that ESP Game produces image tags that tend to be common and uninformative [34, 85]. 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 impose restrictions on what the players are allowed to enter. In the ESP Game, *taboo words* [80] are introduced to prevent players from re-entering high frequency tags. None of these approaches seem to solve the problem completely [85]. 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 [79] forces players to enter only an answer that begin with a particular letter, without knowing that 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. Finally, in designing a game that is fun to play, it is important that the task is neither too easy nor too difficult; depending on the characteristics of the input data, this sometimes necessitates the invention of new mechanisms to make the game enjoyable and effective in collecting high quality data.

There are many other games that do not use the output-agreement mechanism. Verbosity [74], Peekaboom [84] and Phetch [83] are three examples. In Verbosity [74], one player (the *describer*) is given a secret word (e.g., "crown") which he has to 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. Peekaboom [84] is a game for locating objects in images involving two players – one player (the *boomer*) is given an image and a secret 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*). Both players are rewarded when the peeker guesses the correct secret word. In Phetch [83], one player (the *describer*) gives a detailed description of an image (e.g., in the form of a sentence), and a set of *seekers* compete to find the image that the describer holds by searching in a database. In all these cases, the game mechanisms requires one of the players to compute an auxiliary function "what is the secret input object your partner holds, given his or her descriptions of that object?" The outputs of the players are "trusted" if this auxiliary function is computed correctly. I call these types of games *function computation games*.

2.3 Human-in-the-Loop Learning Systems

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 fail-safe manner [28, 67].

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 [50, 70], feature values [54], reward signals [75], target controlled policies [3] and information about hidden states [39]. 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 [17, 72].

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) [38] or perform its own tasks (e.g., finding a room) [67]. 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 [35]. 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.

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 [70]. 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) [50], are disagreed upon by the most experts (query-by-committee) [71], or have the most effects on improving the generalization power of the classifier (Expected Error Reduction) [69] 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.

The single perfect oracle assumption

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 [20], which examines instance selection strategies when there are many imperfect oracles. For example, Pinar et al. [19] studied the problem of how to make as few as 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 [19]) 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* [59] 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 [21]. 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 [36, 37] 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 $(1 - \alpha)$ 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 [21], 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 [22] 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 [23], 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 [87].

The closed world assumptions

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 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 [18, 53] 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 [1]:

> "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 asks 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.

3 Thesis Proposal

Starting with some basic definitions, I will clarify the terminologies used in this thesis. I will then outline the hypotheses in the *how* and *what* aspects of our system, describe the optional work on the *who* aspect, and introduce ELF, a prototype information retrieval system for showcasing the utility of the attributes extracted by our system.

3.1 Definitions

The definitions below distinguish the *attributes* of an object from its *features*, *tags* and *classes*. In particular, I hold the point of view that the notion of class and attribute are equivalent – "is a dog" is considered an attribute, but can be referred to as a class if it is the function that the system is currently trying to learn.

Definition An *object* o is a physical entity, such as a music clip, an image, or a named entity (e.g., Paris), that can be represented as a tuple $\langle \mathcal{F}, \mathcal{A} \rangle$, where $\mathcal{F}$ is a set of low-level features and $\mathcal{A}$ is a set of attributes.

Definition A *feature* f is the low-level machine representation of an object which is automatically extracted, e.g., audio features, color histogram, context pattern co-occurrence counts.

Definition An *attribute* a is a function which maps an object o to an *attribute value* v, i.e., $a : O \rightarrow V$. For example, *isRed*: apple $\rightarrow 1$.

Definition An *tag* is the tuple $\langle a, o, v \rangle$, where a is an attribute and v is the the attribute value for the object o.

Definition An *class* c is an attribute that the system is currently interested in learning.

3.2 *How*: New Game Mechanisms for Extracting Attributes and Attribute Values

3.2.1 Motivation

Because of the limitations of the output-agreement mechanism, it is necessary to invent new game mechanism in order to collect detailed, descriptive semantic attributes that can be used to make fine-grained distinctions between different categories and objects. We introduce a class of games called **set-based function computation games**, where players are given a set of objects for which they need to exchange some information with each other, in order to compute an auxiliary function successfully and receive rewards. In particular, we introduce two new game mechanisms – the **input-agreement mechanism**, where players are given a pair of objects and asked to compute the identity function (i.e., whether the objects are the same or not), and the **complementary-agreement mechanism**, where players are given a set of objects and asked to compute a partition (non-overlapping parts) of the set such that each partition is associated with a different attribute value.

There are a few interesting properties of function computation games involving a *set* of input objects, which we can leverage for the purpose of attribute learning. First, because of open communication, these human computation games can serve as platforms for evaluating machine learning algorithms, by having algorithms pose as game players. This has important implications – attribute learning algorithms can now obtain continuous evaluative human feedback in an economic way, i.e., they can both collect new information *and* monitor their own progress using human supervision. Second, these games can produce explicit positive (e.g., "soothing") and negative examples (e.g., "not soothing") for an attribute. This helps to combat the closed world assumptions. Third, there is evidence that the similarity of the set of objects presented to the players has some effects on players' annotations and how successful they are at completing the task. By choosing the set of objects based on different extent of similarity, we can actually modulate the difficulty of the task, and gather attributes at different levels of granularity. For example, in the input-agreement mechanism, the more similar the pair of objects, the more tags players must exchange in order to correctly guess whether the objects are the same or not. Likewise, in the complementary-agreement mechanism, by presenting clusters of objects of increasing similarity, players would need to pinpoint an attribute that makes finer-grained distinctions amongst categories. Not surprisingly, our results show that similarity is correlated with task difficulty, i.e., players make more mistakes in attempting to distinguish between highly similar objects.

Below are the four desired properties of a human computation games for attribute learning. The first two properties are concerned with player satisfaction and the quality and quantity of the extracted attributes. The last two properties looks at whether the game mechanism, and the attributes extracted by it, are actually useful for classification and retrieval.

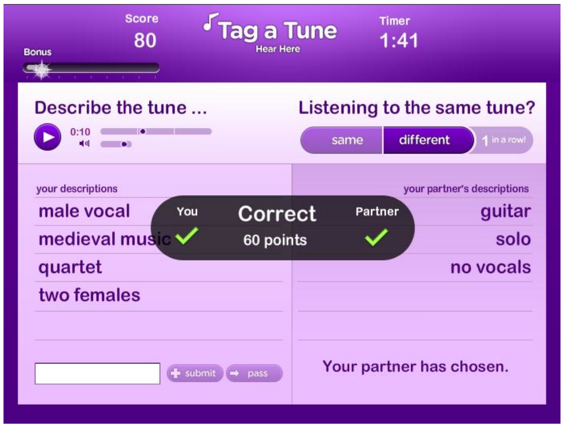
For both the input-agreement mechanism and the complementary-agreement mechanism, we will evaluate whether they have these desired properties. The four desired properties are:

1. **Effectiveness** — Are players successful at completely the task?
Do players like the game?
Is cheating minimized?
2. **Quantity and Quality** — Is the game successful at extracting lots of accurate attributes?
Can the game explicitly collect both positive and negative examples?
3. **Classification and Retrieval** — Are the extracted attributes useful for classification and retrieval?
4. **Evaluation** — Can the game be used to evaluate attribute learning algorithms?

3.2.2 Proposed Work

I. Input-Agreement Mechanism (Completed)

The output-agreement mechanism, such as that introduced by the ESP game, relies on the fact that it is relatively easy to match on a tag. This, we argue, is because there are fewer ways to *name* objects (e.g., “cow”, “person”, “road”) than to describe the *attributes* of objects. For example, while it may be easy for two people to identify the instrumentation (e.g., “violin”, “guitar”) in a piece of music, it is much harder for them to come up with a matching tag to describe its abstract properties, such as temperature (e.g., chilly, warm), mood (e.g., dark, angry, mysterious), or the image it evokes (e.g., busy streets, festival), as well as categorizations that have no clearly defined boundaries (e.g., acid-jazz, jazz-funk, smooth jazz). The difficulty with arbitrary sound clips is even more marked, since the content is not always readily recognizable. Players can enter two attributes that are equivalent in meaning (e.g., “cars”, “traffic on the street”) that fail to match.



Describe the tune ...

Listening to the same tune?

male vocal
medieval music
quartet
two females

guitar
solo
no vocals

Correct
60 points

Your partner has chosen.

Figure 3: Tagatune

We address this challenges in the context of music annotation, and using a game called TagATune (as depicted in Figure 3). In this game, players were each given a piece of music, and the function they are trying to compute is whether the two pieces of music are the same or different. The mechanism behind TagATune leverages the fact that players must be truthful to one another in order to succeed in determining if the music clips are the same or different. The human computation system can then collect accurate descriptions of music by “eavesdropping” on the players’ conversations. This mechanism is called *input-agreement* [81], since players must figure out whether the input objects are the same or not. Like the ESP Game, the input-agreement mechanism is general (i.e., can be applied to annotate other types of data, e.g., images, video, or text) and is particularly useful for input data that has high description entropy (i.e., can be described in many semantically equivalent ways) [47].

1. Effectiveness

Hypothesis How.1a (Completed)

Claim The input-agreement mechanism is effective at extracting attributes and attribute values.

Approach Evaluate game statistics to see if players are successful at completing the task, if they like the game, and if cheating is minimized.

Results: The percentage of rounds that players opted to pass dropped from 36% [48] to 0.5% [47], indicating that the input-agreement mechanism managed to decrease player frustration substantially. Since the system holds the ground truth, there is no easy strategies for players to cheat even though open communication is allowed. Over the course of four months, a total of 26,625 unique games of TagATune were played by 7,893 unique players, equaling 232,804 normal rounds. The number of games each person played ranged from 1 to 948, and the total time each person spent in game play ranged from 3 minutes to 80 hours.

2. Quality and Quantity

Hypothesis How.1b (Completed)

Claim The input-agreement mechanism is successful at extracting lots of accurate attributes.

Approach Analyze the properties, quality and quantity of the collected attributes.

Results: Over the course of four months, we collected a total of 299,003 tags collected on 22,094 audio clips, of which 58,204 were of high confidence (verified by more than two players) [47]. By 7 months, over 1,088,292 tags were collected. We have evaluated the accuracy of the tags by randomly selecting 20 songs and having 80 Mechanical Turk workers evaluate the *uncleaned* tags that were extracted for each song. On average, only about 1 out of 7 tags was considered inaccurate. Another evidence that the attributes extracted by our game are accurate is that when humans play against the aggregate bot, which serves tags from previous games, 93% of the time, humans are able to guess correctly whether the songs are the same or different based on the tags. The attributes collected by TagATune can be detailed, descriptive and sometimes imaginative (e.g., “cookie monster vocal”, “gypsy music”, “80’s pop”, “chinese garden”, “vampires at a dinner party”), and can be extremely noisy (i.e., containing many synonyms, mis-spelled words, and conversational exchanges between players that have nothing to do with the content of the music). TagATune is also able to collect *negation tags*, which are explicit negative examples for an attribute, 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 product of the input-agreement mechanism, and are not often found in output-agreement games where communication is forbidden.

3. Classification and Retrieval

Hypothesis How.1c (Completed)

Claim The input-agreement mechanism can collect attributes that are useful for classification and retrieval.

Approach Train a variety of attribute learning algorithms using the attributes extracted by TagATune.

Results: A large dataset (called MagnaTagatune), containing a *cleaned* set of attributes extracted by TagATune, has been released the research community, and was since used to train a variety of music tagging algorithms. In our own work [46], we have shown that even with the *uncleaned* attributes collected by TagATune as training data, using an approach based on topic models, it is still possible to train reasonable classifiers for learning music attributes.

4. Evaluation

Hypothesis How.1d (Completed)

Claim The input-agreement mechanism can be used to evaluate attribute learning algorithms.

Approach In an off-season MIREX "Special TagATune Evaluation" benchmarking competition, we solicited submissions of music tagging algorithms trained using the MagnaTagatune dataset, and have the algorithms pose as game players in TagATune. Algorithms performance is measured by how well human players can play TagATune against individual algorithms as their game partners.

Results: One very useful side effects of open communication games is that they can be used to evaluate the performance of algorithms. We have investigated with Tagatune, by having music tagging algorithms pose as game players to play against human players. During an evaluation round, the game shows the same piece of music to both the algorithm bot and the human player, and the algorithm bot is asked to generate tags for the music clip. Based on the algorithm's tags, the human player must decide whether the music clips are the same or not. The outputs of the algorithms for each music clip is evaluated by 10 players. There are 100 test songs in total. The algorithm performance is measured by the percentage of people who can guess correctly that the music clips are the same. 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.

Our results show that human players can correctly guess that the music are the same 93% of the times when paired against the aggregate bot (which serves tags emitted by human players in previous games), while only approximately 70% of the times when paired against an algorithm bot, and 28% of the times when paired against a random bot. During this competition, 5,000 human evaluations are collected from 650 players. There is one major weakness in using the input-agreement game mechanism for evaluation. The game experience can be ruined by an algorithm that generates tags 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. In the benchmarking competition, we have systematically removed tags generated by the algorithms that are contradictory or redundant, which reduced but did not eliminate the problem entirely.

II. Complementary-Agreement Mechanism (On-going)

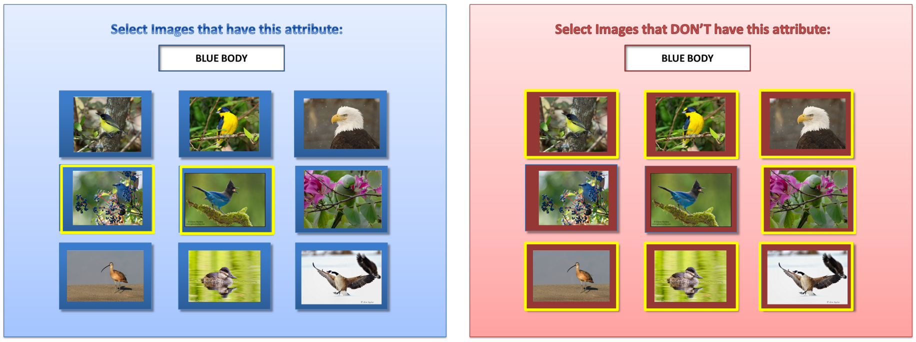
Consider an alternative game for collecting new attribute and attribute values – two players are presented with a set of objects and an attribute and asked to click on the objects that have that attribute. The output-agreement mechanism, in this case, would work poorly – if players are rewarded for agreement, then there is a simple cheating strategy where players click on everything and receive the maximum reward.

To solve this problem, we introduce a new game mechanism called *complementary-agreement* mechanism, where one of the players is asked to generate outputs that the other player are forbidden to enter. Polarity (Figure 4) a is game that implements this mechanism. In this game, two players are presented with a set of objects (e.g., images or named entities), and asked to alternate between two roles – the "positive" player (Figure 4(a)) must select objects that have a given attribute, while the "negative player" (Figure 4(b)) must select objects that do not have an attribute. There are two different modes for a Polarity round: in the *attribute acquisition mode*, the attribute is specified by one of the players; in the *attribute verification mode*, the attribute is specified by the system. Players receive a joint score of $(|\mathcal{S}_p| \times |\mathcal{S}_n|) - c \cdot |\mathcal{S}_p \cap \mathcal{S}_n|$, where $\mathcal{S}_p$ is the set of objects selected by the positive player, $\mathcal{S}_n$ is the number of objects selected by the negative player, and c is the penalty for selections that overlap between the two players. A game round should not end until all objects have been selected by at least one player.

The complementary-agreement mechanism has some interesting properties. First, in a single round of the game, we are able to gather both the positive and negative examples of a given attribute. This allows rapid creation of datasets for training attribute classifiers. Second, since the entire set of the objects are revealed to the players (as opposed to TagATune, where each player is given only one of the objects), we can gather attributes that *explicitly* distinguish between objects that are confusable. Finally, the game allows machine learners to propose new attributes and attribute values to be evaluated by the human players.

The closest task in Psychology is a unsupervised categorization task called free sorting [5] where subjects are given a set of items and are asked to freely sort items to groups. For example, the free sorting experiments in [26, 65] involve a set of objects (e.g., schematic faces) that are synthesized using a small set of attribute and attribute values. It was found that in free sorting tasks, people tend to sort examples using a single dimension [32, 55] using a salient or defining attribute. Subsequently, Ahn and Medin [4] proposes a model where people use a two stage process for categorization, involving first selecting a salient attribute (e.g., size) that describes some of the objects, then grouping the objects by the extreme values (e.g., small versus large) of that attribute, then assigning the rest of the objects (which lack extreme attribute values) to groups by similarity. Along the same lines, in the Polarity, we ask players (or our system) to explicitly name the salient attribute, and then sort the set of objects into two groups.

Extracting Visual Attributes


Select Images that have this attribute:
BLUE BODY
Select Images that DON'T have this attribute:
BLUE BODY

(a) "Positive Player" (b) "Negative Player"

Figure 4: Polarity

I plan to study Polarity within the context of image classification (see Hypothesis How.2a, How.2b, How.2c and How.2d on the following page). As mentioned before, there has been a series of work in Computer Vision attempting to use a *manually curated* intermediate layer of attributes for classification. Our goal is to compare our approach of

1. Effectiveness

Hypothesis How.2a

Claim The complementary-agreement mechanism is effective at extracting attributes and attribute values of images.

Approach Evaluate game statistics to see if players are successful at completing the task, if they like the game, and if cheating is minimized.

2. Quality and Quantity

Hypothesis How.2b

Claim The complementary-agreement mechanism is successful at extracting lots of accurate image attributes.

Approach Run an offline experiment. Use a naive policy for choosing the set of objects to present to players, e.g., randomly select two classes, and then randomly select a set of examples from each class. Analyze the properties, quality and quantity of the collected attributes.

3. Classification and Retrieval

Hypothesis How.2c

Claim The complementary-agreement mechanism can collect attributes that are useful for image classification and retrieval.

Approach Train a two-level classifier using the attributes collected by Polarity. Measure the classification and retrieval performance, and compare the results against previous works, which use a set of manually curated attributes.

4. Evaluation

Hypothesis How.2d

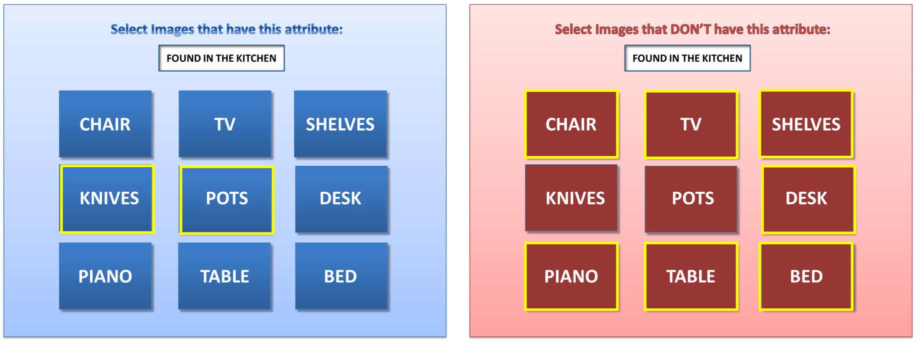
Claim The complementary-agreement mechanism can be used to evaluate image attribute learning algorithms.

Approach Measure whether the evaluative feedback are accurate.

extracting these attributes automatically, in terms of correctness, classification accuracy and retrieval performance, to other work who has used an expert created set of attributes. These datasets include

1. Animals with Attributes [44]: 50 animal classes and 85 attributes.
2. PubFig [42]: 200 person classes and 65 attributes.
3. aPascal [25]: 20 object classes and 64 attributes.
4. CUB-200 [86]: 200 bird classes and 288 binary attributes.
5. LeafSnap [2]

Extracting Non-Visual Attributes



(a) “Positive Player”

(b) “Negative Player”

Figure 5: Polarity (text version)

NELL (which stands for Never-Ending Language Learner) is a web mining system that runs 24 x 7 extracting instances of categories (e.g., Chicago is a city) and relations (e.g., Chicago is in Illinois), by coupling the simultaneously learning of many classifiers. I plan to study how the complementary-agreement mechanism can provide useful human feedback for NELL (see Hypothesis How.3a, How.3b, How.3c, How.3d). In particular, Polarity can be expected to provide two types of human feedback. It can provide corrective feedback:

- **detect wrong instances of categories and relations**: being able to detect characteristically different clusters of named entities amongst the promoted instances of a category is one way to detect errors. For example, the category *ski area* has a set of *rivers* incorrectly promoted as instances. Using the Polarity game, we can have humans define the attributes that distinguish between ski resorts and rivers. Another type of errors that can potentially be detected by finding clusters is *concept drift*, e.g., for the *baked goods* category, amongst other pastries, a set of internet-related named entities are extracted due to the prevalance of the phrase “internet cookies” on the Web.
- **detect ambiguous instances**: some named entities are semantically ambiguous [58], i.e., they can belong to multiple categories. For example, the named entity “Washington” can simultaneously refer to a city, state, person and brand of apple. When NELL classifies an instance into multiple categories, it is difficult to know if the classification is incorrect, or simply ambiguous. Using Polarity, we might be able to detect these semantically ambiguous instances – for example, we can present Washington together with clusters of states and apple names, and observe the number of times that players group Washington into either clusters.
- **detect synonyms**: many concepts can be referred to by multiple names [58]. For example, “intel corp,” “intel corps.,” “intel corporation,” “intel” can all refer to Intel, the company. With Polarity, we can potentially detect these synonyms. For example, we can present players with a set of synonyms and non-synonyms, and provide the attribute “refers to the same thing”.

1. Effectiveness

Hypothesis How.3a

Claim The complementary-agreement mechanism is effective at extracting attributes and attribute values of named entities.

Approach Evaluate game statistics to see if players are successful at completing the task, if they like the game, and if cheating is minimized.

2. Quality and Quantity

Hypothesis How.3b

Claim The complementary-agreement mechanism is successful at extracting lots of accurate attributes for named entities.

Approach Run an offline experiment. Use a naive policy for choosing the set of objects to present to players, e.g., randomly select two classes, and then randomly select a set of examples from each class. Analyze the properties, quality and quantity of the collected attributes. Specifically, evaluate whether Polarity is useful for detecting (i) incorrectly promoted instances, (ii) ambiguous instances, (iii) synonyms, (iv) new subclasses, (v) new relations.

3. Classification and Retrieval

Hypothesis How.3c

Claim The complementary-agreement mechanism can collect attributes that are useful for named entity classification.

Approach Run an offline experiment. Inject different types of human feedback into NELL at iteration X, and contrast the performance of NELL at iteration $X + C$ with or without human feedback.

4. Evaluation

Hypothesis How.3d

Claim The complementary-agreement mechanism can be used to evaluate NELL's beliefs.

Approach Measure whether the evaluative feedback are accurate.

Polarity can also be used to extend NELL's ontology, by either having humans suggest new subclasses and relations, or have NELL propose new subclasses and relations, and have instances of the proposed subclasses or relations verified by humans:

- **discover new subclasses**: For example, different types of athletes are discussed in very different ways on the Web, i.e., the context patterns associated with them may be characteristically different. Therefore, it may be beneficial to have in the ontology categories of "football players", "soccer players", "baseball players" instead of the generic category "athletes". Using Polarity, we can present players with instances from the category "athlete" and potentially obtain "football" as an attribute which divides the instances into football versus non-football players.
- **discover new relations**: For example, ("cups", "kitchen") and ("chairs", "living room") are two instances of the relation *isFoundIn* that we can discover using the Polarity game (see the example in Figure 5).

3.3 *What*: Active Learning for Attribute Acquisition

3.3.1 Motivation

In the experiments we have described so far, the way that the set of objects are chosen in Polarity is naive. I plan to investigate more sophisticated ways of selecting objects to present to players, in order to efficiently achieve the objectives of the learning systems. I propose to investigate the **active attribute acquisition** problem both within the context of image classification and category and relation learning in NELL.

3.3.2 Proposed Work

I. Image Classification

Hypothesis What.1	
Claim	Our active learning strategy outperforms the random baseline, in terms of image classification performance.
Approach	Compare various active learning strategies for attribute acquisition against the random baseline. Evaluate by measuring classification accuracy against the number of queries (a query being a round of the game, in this case).

I propose to investigate an active learning strategy for selecting the set of objects to present to players in order to obtain attribute and attribute values for the purpose of classification. In particular, the learning problem is two-layer classification problem. The learner is given a set of images $\mathcal{O} = \{o_1, \ldots, o_N\}$, where each object o_i is represented as a tuple $(\mathcal{F}_i, \mathcal{A}_i)$, where $\mathcal{F}_i$ is a set of low-level features (e.g., color histogram) and $\mathcal{A}_i$ is a set of attributes, which is $\emptyset$ at the beginning. Supposed that the learner is also given a set of target attributes $\mathcal{Y}$, which we can also refer to as *classes*, to learn. The goal is to obtain from the human oracles (through the Polarity game) the attribute and attribute values for each object in $\mathcal{O}$, in order to learn to classify new objects (i) by the high-level attributes $f_1 : A \rightarrow Y$, and (ii) by low-level attributes via an intermediate attribute layer, $f_2 : X \rightarrow Y$ where $f_2 = g_1 \circ g_2$, where $g_1 : X \rightarrow A$ and $g_2 : A \rightarrow Y$.

Intuitively, the active learning algorithm for attribute acquisition can choose examples to present to humans in order to (i) better disambiguate between confusable classes by asking for a new attribute (and attribute values) that distinguish them, or (ii) improve one of the attribute classifiers, by choosing examples that are the most ambiguous (e.g., have 0.5 probability of having that attribute). Consider the simple algorithm below, which will serve as the basic skeleton of our algorithm.

Algorithm 1 Active Attribute Acquisition for Image Classification

```
Repeat
    1. Train a two-level classifier $f_2 = g_1 \circ g_2$.
    2. Decide whether to
        (a) acquire a new attribute
            Select $\mathcal{O} = \{o_1, \ldots, o_K\}$
        (b) acquire attribute values for an existing attribute
            Select an existing attribute $a$
            Select $\mathcal{O} = \{o_1, \ldots, o_K\}$
    3. Train the attribute classifiers using the collected attribute and attribute values.
    4. Use the trained attribute classifier to update the attribute value prediction for all images.
```

The work of Parikh et al. [62] is closest to ours. They are also concerned with the classification of objects by acquiring new attributes from workers on Mechanical Turk. However, their query type is different – they present a set of images that belong to the opposite sides of an attribute hyperplane (e.g., with the least "cloudy" images on the left and the most “cloudy” images on the right), and ask the human worker whether there exist an attribute of the images (e.g., cloudiness) that is increasing from left to right, and name that attribute when possible. To construct a query to the oracle, their algorithm (i) selects examples $\mathcal{E}_c$ from two or more *confusable* classes, (ii) uses unsupervised iterative max-margin clustering to find two clusters in $\mathcal{E}_c$, assign all training examples to one of the two clusters, then build an SVM classifier to discriminate between these two clusters, and (iii) uses a mixture of probabilistic principal component analyzers (MPPCA) to predict the *nameability* of the attribute, which they define as the probability that the projected SVM hyperplane parameters are generated by the mixture of subspaces. Their work has only demonstrated utility on extracting global attributes (e.g., “green”, “man-made”); the applicability of their approach to finding more specific attributes that distinguish bewteen fine-grained categories is yet unknown.

II. Category and Relation Learning

Hypothesis What.2	
Claim	Our active learning strategy outperforms the random baseline, in terms of precision and recall of the category and relation extractions in NELL.
Approach	Compare various active learning strategies for attribute acquisition against the random baseline. Evaluate by measuring precision/recall against the number of queries (a query being a round of the game, in this case).

An equivalent problem exists in NELL – our active learner must decide in each iteration, whether to acquire a new category or relation in order to extend the ontology, or to verify the membership of instances of an existing category and relation in order to correct the ontology. The solution is potentially a decision theoretic algorithm, which will make this decision depending on the extent to which different types of feedback is projected to help improve NELL in the next iterations. Consider the simple algorithm below, which will serve as the basic skeleton of our algorithm.

Algorithm 2 Active Attribute Acquisition for Category and Relation Learning in NELL

```
Repeat
    1. Train NELL
    2. Decide whether to
        (a) extend current ontology
            Select $\mathcal{O} = \{o_1, \ldots, o_K\}$
        (b) correct current ontology
            Select an existing class or relation $a$
            Select $\mathcal{O} = \{o_1, \ldots, o_K\}$
    3. Train the category or relation classifiers using the collected attribute and attribute values.
    4. Use the trained category or relation classifier to update the category or relation predictions for all named entities.
```

We expect there to be some synergies between the two active learning algorithms we propose here for image classification and category/relation learning, which we will explore in this thesis.

3.4 *Who*: Taking Humans Into Account (Optional)

3.4.1 Motivation

In the perfect world, one can simply use information theoretic ways to select input objects to greedily achieve the system's objectives. This is the assumption made in the *what* section, where we consider different strategies for selecting objects to present to players. However, humans have different levels of competence and expertise that need to be accounted for. As **possible extensions to this thesis**, I propose to run a few smaller studies to investigate the *who* aspect of human computation systems, in particular answering two questions: (1) How does task difficulty, modulated by the similarity of the set of objects we ask players to discriminate, affect players' ability to succeed at the task (i.e., score) and the amount of effort (i.e., time spent) it takes for them to succeed? (2) How does expertise affect the quality and characteristics of the attributes and attribute values obtained by our system?

3.4.2 Proposed Work

I. Modeling General Competence

Hypothesis Who.1

Claim By employing a task routing strategy which adjusts to the player's changing level of competence, we can keep the game neither too difficult or too easy for players.

Approach Compare three conditions – easy (root node classification), difficult (leaf node classification) and adaptive level of difficulty (e.g., by dynamically changing the level depending on each player's current score and effort level). Measure the scores and effort levels (in terms of time) of the players in each condition.

Part of what makes human computation games fun is the fact that players feel a sense of accomplishment succeeding at a task. It is therefore not ideal to ask novice players to do difficult discrimination tasks, or even expert players to *always* do difficult discrimination task. We need to balance for difficulty and adjust *what* we select for each individual worker according to their current score or effort level (in terms of time spent solving each round) in the game. This is similar to adaptive testing [6, 68], where test items are selected to tailor to the abilities of the examinees. If an examinee performs well on an item at a certain level of difficulty, he will be presented with a more difficult question, vice versa.

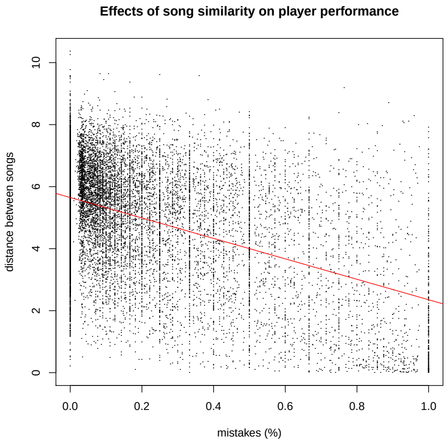


Figure 6: Tagatune

Figure 7: The correlation of player performance with music similarity

Consider the observation of player performance in TagATune in Figure 6. The graph shows that the more similar the music (i.e., the lesser the distance), the more mistakes players made in guessing that the music clips are the same when in fact they are different. Here, the distance between two music clips is measured using the KL divergence between the topic distributions (computed using the collected tags) of the pair of songs. This implies that similarity is related to task difficulty. I propose to further evaluate this hypothesis using the Polarity game, and investigate ways of assigning tasks (i.e., sets of objects) to players such that their success rate or time spent is not overly divergent from the average. The problem of inferring task difficulty can be simplified if we assume that the ontology dictates the difficulty of the task. For example, one simplifying assumption is that the deeper into the ontology, the more similar the objects, the harder the task is (i.e., players will succeed less often) and the longer it will take players to perform the task successfully. Hypothesis Who.1 outlines a user study for investigating this question.

II. The Role of Expertise

Hypothesis Who.2

Claim Polarity will generate characteristically different attributes depending on the expertise level of the players.

Approach Run a user study with three conditions: (a) expert against expert, (b) expert against novice, (c) novice against novice. The domains can be restricted to medicine (for text-based Polarity) and birds (for image-based Polarity). Report the qualitative and quantitative differences between the attribute and attribute values extracted, as well as the players' satisfaction with the game experience, in each of the three conditions.

Especially in the situation when the input objects are named entities, expertise is expected to play an important role. For example, medical experts who are given names of diseases are more likely to be able to immediately pick out the interesting commonalities between the diseases. I plan to run a small study (see Hypothesis Who.2), to understand how the feedback in Polarity might differ if the players are domain experts, versus the general population. This may point out the challenge of using Polarity on named entities – unlike most human computation games, where the task can be easily accomplish by any players with no special knowledge or abilities, in our case, knowledge might be a necessary pre-requisite in order for players to succeed in the game (thereby, enjoy the experience) and produce useful results. I want to evaluate the extent of this limitation, and interface supports (e.g., hints, ability for players to choose a domain, e.g., "sports", that they know a lot about or are interested in [45]) that can be used to bridge this gap.

3.5 Application

3.5.1 ELF

Hypothesis App.1

Claim Polarity produces a set of attribute and attribute values that can support a variety of retrieval and identification tasks.

Approach Build a prototype version of ELF to showcase the utility of the attributes extracted by our system. Measure how successful users are in performing different retrieval and identification tasks using the lookup and finder modes of ELF.

We will showcase the learned attributes of our human computation system by building a **prototype** information retrieval system called ELF (Entity Lookup-Finder) which allows users to search for and identify objects using both visual and non-visual attributes, extracted by our system. ELF operates under two modes – the lookup mode and the finder mode. In the **lookup** mode, users are presented with a set of attributes which they can use to assemble a complex query, such as "has red beak, round body, green spot on the wings," and obtain a set of objects that match those attributes. In the **finder** mode, ELF can play a 20 question-like game with the user, by presenting the user with a set of questions regarding the attributes of the object he or she is looking for, and using the responses of the user to retrieve a set of candidate objects.

In Section 1.1, we have previously described several retrieval and identification scenarios, e.g., naming plants, identifying people, discovering birds, locating objects. I propose to develop ELF to handle some of those scenarios, in order to demonstrate the utility of the attributes learned by our system.

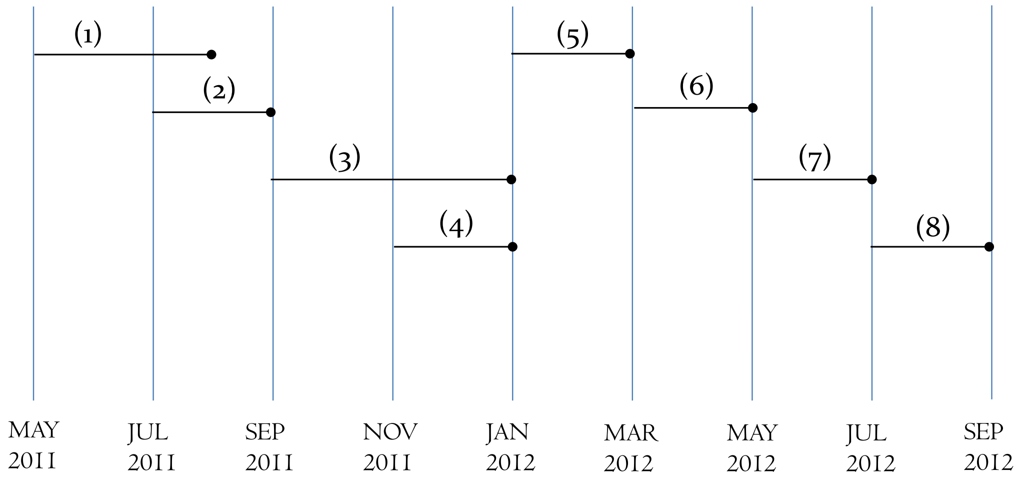
4 Summary

Focusing on attribute learning, this thesis investigates research issues surrounding the building of a large-scale machine-in-the-loop human computation system, using games as a platform. By testing a set of focused hypothesis, we will explore *how* to extract attributes and attribute values from humans using games and *what* active learning strategies are appropriate for attribute acquisition. In addition, this thesis will contribute a set of practical platforms and datasets for advancing the study of attribute learning for music, images and named entities.

The work described in this thesis proposal can be viewed also as a case study of the role of machine intelligence in human computation systems. Human computation systems that require coordination of large number of individuals, e.g., to collaborative solving planning problems, can benefit enormously from the incorporation of AI and machine learning algorithms. Two of my on-going projects – CrowdSearch and Mobi – are beginning to probe at the question of how to leverage both human and machine intelligence at solving large-scale collaborative planning problem. It is the hope that these projects, as well as the work presented in this thesis, will contribute deeper understanding on the potentials and challenges associated with machines-in-the-loop human computation systems.

5 Schedule of Work

We detail here the expected deliverables and a timeline for achieving them.



(1) Deploy Polarity, Hypothesis How.2a, 2b, 3a, 3b

(2) Hypothesis How 2c, 2d, 3c, 3d

(3) Hypothesis What.1, What.2

(4) Prepare Job Applications

(5) Hypothesis What.2

(6) Develop ELF

(7) Hypothesis App.1

(8) Thesis Writing

6 References

[1] Artificial intelligence in the open world. 2.3.1

[2] Leafsnap website. `http://leafsnap.com`. 5

[3] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement. In *ICML*, 2004. 2.3

[4] W. Ahn and D. Medin. A two-stage model of category construction. 16:81–121, 1992. 3.2.2

[5] F. G. Ashby and W. T. Maddox. *Stimulus Categorization*, pages 251–301. Academic Press, 1998. 3.2.2

[6] F. Baker. *The Basics of Item Response Theory*. Heinemann, 1985. 3.4.2

[7] M. Banko and O. Etzioni. Strategies for lifelong knowledge extraction from the web. In *K-CAP*, 2007. 2.1.3

[8] T. Berg, A. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. In *ECCV*, 2010. 2.1.2

[9] J. Bergstra, A. Lacoste, and D. Eck. Predicting genre labels for artists using freedb. In *ISMIR*, pages 85–88, 2006. 2.1.1

[10] 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*, 37(2):115–135, 2008. 2.1.1

[11] J. Betteridge, A. Carlson, S. A. Hong, E. R. H. Jr., E. Law, T. Mitchell, and S. H. Wang. Toward never ending language learning. In *Proceedings of AAAI Spring Symposium on Learning by Reading and Learning to Read*, pages 1–2, 2009. `www.cs.cmu.edu/~elaw/aaai.pdf`. 2.1.3

[12] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT)*, pages 92–100, 1998. `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.33.2452`. 2.1.3

[13] A. Carlson, J. Betteridge, E. R. H. Jr., and T. Mitchell. Coupling semi-supervised learning of categories and relations. In *Proceedings of the NAACL–HLT Workshop on Semi-supervised Learning for Natural Language Processing*, pages 1–9, 2009. `http://rtw.ml.cmu.edu/papers/cbl-sslnlp09.pdf`. 2.1.3

[14] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. H. Jr., and T. Mitchell. Towards an architecture for never-ending language learning. In *AAAI*, 2010. 2.1.3

[15] A. Carlson, J. Betteridge, S. Wang, E. Hruschka, and T. Mitchell. Coupled semi-supervised learning for information extraction. In *Submission*, 2009. 2.1.3

[16] O. Chapelle, B. Scholkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, 2006. 2.1.3

[17] A. Culotta, T. Kristjansson, A. McCallum, and P. Viola. Corrective feedback and persistent learning for information extraction. 170(14–15):1101–1122, 2006. 2.3

[18] L. B. D. Turnbull, R. Liu and G. Lanckriet. A game-based approach for collecting semantic annotations for music. In *ISMIR*, pages 535–538, 2007. 2.2.1, 2.3.1

[19] P. Donmez and J. Carbonell. Proactive learning: Cost-sensitive active learning with multiple imperfect oracles. In *CIKM*, 2008. 2.3.1

[20] P. Donmez and J. Carbonell. *From Active to Proactive Learning Methods*. Springer: Studies in Computational Intelligence, 2009. 2.3.1

[21] P. Donmez, J. Carbonell, and J. Schneider. Efficiently leanring the accuracy of labeling sources for selective sampling. In *KDD*, 2009. 2.3.1

[22] P. Donmez, J. Carbonell, and J. Schneider. A probabilistic framework to learn from multiple annotators with time-varying accuracy. In *SDM*, 2010. 2.3.1

[23] P. Donmez, J. Carbonell, and J. Schneider. Probabilistic framework to learn from multiple annotators with time-varying accuracy. In *SIAM*, 2010. 2.3.1

[24] A. Farhadi, I. Endres, and D. Hoiem. Attribute-centric recognition for cross-category generalization. In *CVPR*, 2010. 2.1.2

[25] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009. 2.1.2, 3

[26] C. Frith and U. Frith. Feature selection and classification: A developmental study. 25:413–428, 1978. 3.2.2

[27] S. Hacker and L. von Ahn. Matchin: Eliciting user preferences with an online game. In *submission*. 2.2.1

[28] M. Hearst. Mixed-initiative interaction: Trends and controversies. pages 14–23, 1999. 2.3

[29] M. Hoffman, D. Blei, and P. Cook. Easy as CBA: A simple probabilistic model for tagging music. In *ISMIR*, pages 369–374, 2009. 2.1.1

[30] E. Horvitz and T. Paek. Complementary computing: Policies for transferring callers from dialog systems to human receptionists. 17, 2007. 2.2

[31] U. Hustadt. Do we need the closed-world assumption in knowledge representation. In F. Baader, M. Buchheit, M. Jeusfeld, and W. Nutt, editors, *Knowledge Representation Meets Databases Workshop*, 1994. 2.1.3

[32] S. Imai and W. R. Garner. Discriminability and preference for attributes in free and constrained classification. 69(6):596–608, 1965. 3.2.2

[33] M. O. Jackson. *Mechanism Theory*. EOLSS Publishers, 2003. 2.2.1

[34] S. Jain and D. Parkes. A game-theoretic analysis of games with a purpose. In *WINE '08: Proceedings of the 4th International Workshop on Internet and Network Economics*, pages 342–350, Berlin, Heidelberg, 2008. Springer-Verlag. 2.2.1

[35] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR*, 2005. 2.3

[36] L. Kaelbling. *Learning in Embedded Systems*. The MIT Press, Cambridge, MA, 1993. 2.3.1

[37] L. Kaelbling, M. Littman, and A. Moore. Reinforcement learning: A survey. 4:237–285, 1996. 2.3.1

[38] A. Kapoor and E. Horvitz. Principles of lifelong learning for predictive user modeling. In *UM*, 2007. 2.3

[39] A. Kapoor and E. Horvitz. Experience sampling for building predictive user models: a comparative study. In *CHI*, 2008. 2.3

[40] 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. 2.1.2

[41] Y. Kim, E. Schmidt, and L. Emelle. Moodswings: A collaborative game for music mood label collection. In *ISMIR*, pages 231–236, 2008. 2.2.1

[42] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *IEEE International Conference on Computer Vision (ICCV)*, Oct 2009. 2.1.2, 2

[43] P. Lamere. Social tagging and music information retrieval. *Journal of New Music Research*, 37(2):101–114, 2008. 2.1.1

[44] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009. 2.1.2, 1

[45] E. Law, P. Bennett, and E. Horvitz. The effects of choice in routing relevance judgments. In *SIGIR*, 2011. 3.4.2

[46] E. Law, B. Settles, and T. Mitchell. Learning to tag from open vocabulary labels. In *ECML*, 2010. 3.2.2

[47] E. Law and L. von Ahn. Input-agreement: A new mechanism for data collection using human computation games. In *CHI*, pages 1197–1206, 2009. 3.2.2, 3.2.2, 3.2.2

[48] E. Law, L. von Ahn, R. Dannenberg, and M. Crawford. Tagatune: A game for music and sound annotation. In *ISMIR*, 2007. 3.2.2

[49] B. Lee and L. von Ahn. Squigl: a web game to generate datasets for object detection algorithms. In *submission*. 2.2.1

[50] D. D. Lewis and J. Catlett. Heterogeneous uncertainty sampling for supervised learning. In *ICML*, 1994. 2.3,

2.3.1

[51] T. Li, M. Ogihara, and Q. Li. A comparative study on content-based music genre classification. In *SIGIR*, pages 282–289, 2003. 2.1.1

[52] M. Mandel and D. Ellis. Song-level features and support vector machines for music classification. In *ISMIR*, 2005. 2.1.1

[53] M. Mandel and D. Ellis. A web-based game for collecting music metadata. 37(2):151–165, 2009. 2.2.1, 2.3.1

[54] S. Maytal, P. Melville, and F. Provost. Active feature-value acquisition for model induction. 55(4):664–684, 2009. 2.3

[55] D. L. Medin, W. D. Wattenmaker, and S. E. Hampson. Family resemblance, conceptual cohesiveness, and category construction. 19:242–279, 1987. 3.2.2

[56] A. Mityagin and M. Chickering. Picturethis. `http://club.live.com/Pages/Games/GameList.aspx?game=Picture_This`. 2.2.1

[57] T. P. Mohamed, E. R. H. Jr., and T. M. Mitchell. Discovering relations between noun categories. In *In Submission*, 2011. 2.1.3

[58] T. P. Mohamed, E. R. H. Jr., and T. M. Mitchell. Which noun phrases denote which concepts. In *In Submission*, 2011. 3.2.2

[59] D. North. A tutorial introduction to decision theory. 4(3):200–210, 1968. 2.3.1

[60] D. N. Osherson, J. Stern, O. Wilkie, M. Stob, and E. E. Smith. Default probability. 15(2), 1991. 2.1.2

[61] M. Palatucci, D. Pomerleau, G. Hinton, and T. Mitchell. Zero-shot learning with semantic output codes. In *Neural Information Processing Systems (NIPS)*, December 2009. 2.1.2

[62] D. Parikh. Interactively building a discriminative vocabulary of nameable attributes. In *CVPR*, 2011. 2.1.2, 3.3.2

[63] M. Poesio and A. Almuhareb. Extracting concept descriptions from the web: the importance of attributes and values. In *Conference on Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, 2008. 2.1

[64] Read the web project. `http://rtw.ml.cmu.edu/wsdm10_online/`. 2.1.3

[65] S. K. Reed. Pattern recognition and categorization. 3(3):382–407, 1972. 3.2.2

[66] M. Rohrbach, M. Stark, G. Szarvas, I. Gurevych, and B. Schiele. What helps where – and why? semantic relatedness for knowledge transfer. In *CVPR*, 2010. 2.1.2

[67] S. Rosenthal. *Human Modeling and Interaction for Effective Task Autonomy: A Thesis Proposal*. PhD thesis, Carnegie Mellon University, October 2010. 2.3

[68] E. E. Roskam. *Models for Speed and Time-Limit Tests*. Springer, 1997. 3.4.2

[69] N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. In *ICML*, pages 441–448, 2001. 2.3.1

[70] B. Settles. Active learning literature survey. Technical report, 2009. 2.2, 2.3, 2.3.1

[71] H. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *ACM Workshop on Computational Learning Theory*, pages 287–294, 1992. 2.3.1

[72] M. Shilman, D. Tan, and P. Simard. Cuetip: A mixed-initiative interface for correcting handwriting errors. In *CHI*, pages 323–332, 2006. 2.3

[73] K. Siorpaes and M. Hepp. Games with a purpose for the semantic web. *IEEE Intelligent Systems*, 23(3):50–60, 2008. 2.2.1

[74] R. Speer, C. Havasi, and H. Surana. Using verbosity: Common sense data from games with a purpose. In *FLAIRS*, 2010. 2.2.1

[75] A. L. Thomas and C. Breazeal. Reinforcement learning with human teachers: understanding how people want

to teach robots. In *AAAI*, 2006. 2.3

[76] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Towards musical query-by-semantic description using the CAL500 data set. *SIGIR*, pages 439–446, 2007. 2.1.1

[77] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Five approaches to collecting tags for music. In *ISMIR*, 2008. 2.1.1

[78] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Semantic annotation and retrieval of music and sound effects. *TASLP*, 16(2):467–476, February 2008. 2.1.1

[79] 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*, pages 535–538, 2008. 2.2.1

[80] L. von Ahn and L. Dabbish. Labeling images with a computer game. In *CHI*, pages 319–326, 2004. 2.2.1

[81] L. von Ahn and L. Dabbish. Designing games with a purpose. 51(8), August 2008. 2.2.1, 3.2.2

[82] L. von Ahn and L. Dabbish. General techniques for designing games with a purpose. In *CACM*, pages 58–67, 2008. 2.2

[83] L. von Ahn, S. Ginosar, M. Kedia, R. Liu, and M. Blum. Improving accessibility of the web with a computer game. In *CHI*, 2006. 2.2.1

[84] L. von Ahn, R. Liu, and M. Blum. Peekaboom: A game for locating objects in images. In *CHI Notes*, pages 55–64, 2006. 2.2.1

[85] I. Weber, S. Robertson, and M. Vojnovic. Rethinking the esp game. In *CHI*, pages 3937–3942, 2009. 2.2.1

[86] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010. 4

[87] P. Welinder and P. Perona. Online crowdsourcing: rating annotators and obtaining cost-effective labels. In *CPVR*, 2010. 2.3.1