Datenbank Spektrum (2012) 12:109–120
DOI 10.1007/s13222-012-0092-8

SCHWERPUNKTBEITRAG

Information Extraction Meets Crowdsourcing: A Promising Couple

Christoph Lofi · Joachim Selke · Wolf-Tilo Balke

Received: 10 April 2012 / Accepted: 12 May 2012 / Published online: 23 May 2012

Abstract Recent years brought tremendous advancements in the area of automated information extraction. But still, problem scenarios remain where even state-of-the-art algorithms do not provide a satisfying solution. In these cases, another aspiring recent trend can be exploited to achieve the required extraction quality: explicit crowdsourcing of human intelligence tasks. In this paper, we discuss the synergies between information extraction and crowdsourcing. In particular, we methodically identify and classify the challenges and fallacies that arise when combining both approaches. Furthermore, we argue that for harnessing the full potential of either approach, true hybrid techniques must be considered. To demonstrate this point, we showcase such a hybrid technique, which tightly interweaves information extraction with crowdsourcing and machine learning to vastly surpass the abilities of either technique.

1 Introduction

By effectively bridging the gap between human knowledge and automatic information processing, algorithmic information extraction (IE) has proven itself an indispensable building block of today's information systems. Based on the key idea of transforming semi-structured or natural language sources into machine-processable structured information, IE research has created an extensive toolbox offering a solution for almost any extraction task and data source. An extensive survey of this fascinating research discipline can be found, e.g., in [1, 2], or the overview article also contained in this journal issue.

While information extraction tasks based on factual concepts that have been stated explicitly and follow a clear pattern nowadays can be handled with reasonable performance, there still is considerable room for improvement in other settings. For example, dealing with implicit and only vaguely defined concepts expressed in common language still turns out to be very demanding. Here, the key challenge is to create algorithmic rules that closely resemble intuitive human understanding—a highly complex and time-consuming task, which gradually seems to reach its limits.

To overcome these obstacles, a general trend towards directly tapping the *human side of data* has become apparent recently [3]. In particular, crowdsourcing systems have been identified as an effective tool making human skills and intelligence accessible to machines. More specifically, they exploit the *wisdom of the crowds* [4], the "intelligence" emerging when cleverly combining independent inputs from a large number of individuals.

In general, the term crowdsourcing may be attributed to any system or platform that explicitly or implicitly enlists a vast number of humans to collaboratively solve complex problems [5]. This ranges from explicit human collaboration efforts creating complex artifacts (e.g., Wikipedia or open source software) across sites based on user-generated content (e.g., YouTube) to sites implicitly exploiting human efforts by aggregating user opinions such as ratings or reviews (e.g., Netflix, IMDb, or Amazon.com). Thus, the *Social Web* is also based on an extensive but mostly uncontrolled crowdsourcing effort (and therefore, by extracting

C. Lofi (✉) · J. Selke · W.-T. Balke
Institut für Informationssysteme, Technische Universität Braunschweig, Braunschweig, Germany
e-mail: lofi@ifis.cs.tu-bs.de
url: http://www.ifis.cs.tu-bs.de

J. Selke
e-mail: selke@ifis.cs.tu-bs.de

W.-T. Balke
e-mail: balke@ifis.cs.tu-bs.de

information from the Social Web, there is already a trivial synergy between IE and crowdsourcing).

Each crowdsourcing system has to face four fundamental challenges [5]: How to recruit and retain users? What contributions can users make? How to combine the contributions to solve the target problem? How to evaluate users and their contributions? Overcoming these challenges usually requires extensive effort for creating and carefully nurturing the required user communities. However, this requirement typically limits their flexibility and usefulness for ad-hoc tasks: Most crowdsourcing systems are very specific to their intended task, and cannot easily be re-purposed or restricted without alienating their user community due to the chosen incentive model. Many platforms rely on volunteers who donate their time because they believe in the platform's mission, want to help their peers, or want to establish themselves or even earn some fame in the respective user community; changing the mission may easily drive these motivated volunteers away.

Therefore, keeping information extraction in mind, we will use a much narrower definition of crowdsourcing in the course of this paper: We will only focus on explicit crowdsourcing for general tasks based on controlled task execution as provided by services such as Amazon's Mechanical Turk, CrowdFlower, or SamaSource. Here, a large problem is solved by dividing it into many small and simple tasks (called HITs, Human Intelligence Tasks; the smallest unit of crowdsourceable work), which then are distributed to a human worker pool. Workers are recruited and retained by paying them, and hence, such platforms could theoretically be used to perform any given dividable task that requires human intelligence. These platforms have successfully been used by researchers from many different domains, e.g., databases operating on incomplete data [6, 7], disaster response [8], or general query processing [9].

Therefore, this type of platform seems to be perfectly suited for information extraction: Every time the need for extraction or integration arises, a respective task can be issued to a crowdsourcing platform in an ad-hoc fashion, i.e., even for very difficult extraction tasks (e.g., extracting conceptual information from natural language, audio, or video), the required cognitive power simply can be bought online.

However, there are some variables limiting the feasibility of crowdsourcing for such tasks: Mainly the *quality* of the human task execution, the *time* needed for executing the tasks, and, of course, the resulting monetary *costs*.

The central challenge in employing crowdsourcing alongside (or even instead of) information extraction is controlling these three variables. Unfortunately, this challenge is far from being simple.

Our contribution for the remainder of this paper is as follows:

- We show how to use *straight-forward crowdsourcing tasks* to address typical problems encountered in information extraction.
- We provide a *methodical classification* of the crowdsourcing tasks relevant to information extraction.
- We will identify *influence factors* that negatively or positively affect the result quality, execution time, and costs of these tasks. Also, we will discuss the relationship of these influence factors and the previously identified task classes.
- For each task class, we will present suitable techniques and approaches that can help to overcome time, costs, or quality issues.
- Finally, we will argue that best results can be achieved when crowdsourcing and information extraction is deeply interwoven into a *true hybrid system*, combining the strength of both approaches and avoiding their individual weaknesses. We demonstrate such an approach and show its potential for the task of extracting perceptual characteristics for a large number of movies (e.g., genre classifications) just from a collection of numerical ratings (e.g., "user x rates movie y with z of 5 stars").

2 Crowdsourcing in Information Extraction

In this section, we briefly highlight three problem scenarios where today's information extraction algorithms are severely challenged, and show how crowdsourcing can be used in a straightforward fashion as an ancillary technique.

Scenario 1: Extraction Impossible

In this scenario, the information to be extracted is neither explicitly nor implicitly available in the given data source. But still, it may be found in other data sources that are beyond the scope or reach of the extraction algorithm. For a simple example consider trying to extract basic information on all German computer science professors in tabular form by crawling the respective department sites. If there is no mention of a professor's phone number, it cannot be extracted. A straightforward application of crowdsourcing in this scenario could be to create a HIT for each missing phone number, and require the crowd workers to obtain the number by calling the department's secretary and simply ask for the number. Of course, this type of crowd sourcing will most probably be very costly, as this HIT is comparably complex and needs to be compensated accordingly in order to produce good results. That is, only reliable and non-malicious workers should be used as quality assurance using majority votes is irritating in this scenario, workers need a phone and also have to pay for the call, and workers must be able and willing to verbally converse in German.

Scenario 2: Extraction Too Expensive in Practice
In this problem scenario, the required information is (implicitly) available in the data source and suitable extraction methods exist. However, actually to apply these methods is too costly for the given task at hand. Often, this involves laboriously hand-crafting or training a complex suite of different extraction algorithms.
Consider the previous example: If some web sites to be crawled only contain email addresses that have been encoded as an image (to avoid spam), then in principle these addresses could be extracted by employing OCR techniques. However, the effort for setting up such an extractor might be too high just for compiling a contact list. In this case, crowdsourcing can be a cheap alternative for obtaining the information by issuing HITs for manually extracting the address from the image.

Scenario 3: Extraction Requires Research
Especially implicit information still poses a severe challenge for state-of-the-art information extraction techniques, e.g. information encoded in natural language text, audio, images, or videos. Often, techniques approaching this kind of information heavily rely on heuristics, statistics, or language modeling, thus making result quality a major research issue. As an example, consider the challenge of relation extraction from natural language [1]: Here, relationships between entities have to be extracted, e.g., as RDF triples from the textual descriptions of Wikipedia pages. While there are many proposed solutions for this problem, e.g., [10–13], extraction quality is still a major issue.
Here, crowdsourcing is not as straightforward as in the previous cases. Early research on this issue, e.g., Cimple/DBLife [14–16], automatically extracts data from the web into structured data. This data is then converted to structured wiki pages, and users are asked to correct and augment the extracted data. However, these approaches have to invest heavily into building the required communities, and lack the flexibility to be used for arbitrary ad-hoc tasks, therefore using general task-based crowdsourcing might be a better solution when the challenges described in the next section can be overcome.

3 Challenges of Crowdsourcing

While crowdsourcing is an extremely powerful emerging technology, applying it naively does often not yield satisfying results. Three major characteristics can be identified, which quickly can become problematic [7]: the *quality* of the human input, the *time* needed for executing tasks, and of course, the resulting *costs*. In the following, we will briefly cover each characteristic and point out those influence factors having a negative impact.

3.1 Answer/Solution Quality

The quality of workers available to crowdsourcing platforms is hard to control, thus making elaborative quality management necessary [17]. This usually requires executing each HIT multiple times, further increasing the costs of each crowdsourcing task, and also increasing the required execution time. Especially HITs with unclear results need to be performed more often in order to be able to rely on majority votes. Unfortunately, usually it is hard to determine upfront how many times each particular HIT needs to be assigned for a reliable result. Therefore, adaptive balancing of quality and performance remains an open issue.

Poor worker result quality can be attributed to two effects: (a) insufficient worker *qualification* (i.e., workers lack the required competencies to solve the presented task) and (b) worker *maliciousness* (i.e., workers do not honestly perform the issued task). Especially, maliciousness is a severe challenge to crowd sourcing: As workers are paid for each solved task, a significant percentage of the general worker population of platforms such as Amazon Mechanical Turk aims at improving their personal income by cheating. This effect is further showcased and evaluated in the next section.

3.2 Execution Time

It has been shown that each requester in a crowdsourcing platform can only utilize a relatively small human worker pool [6]. This means, that HITs cannot be parallelized arbitrarily as the number of simultaneous human computations is capped by the worker pool size, thus the scalability of a system relying on such platforms is potentially hampered. While HITs can carry out semantically powerful operations, completing large HIT groups may take very long and impose severe performance challenges [18].

Furthermore, the attractiveness of HITs further influence the time needed for their execution. The attractiveness can be positively influenced by increasing payment [6], and by decreasing the required effort (or perceived task complexity).

3.3 Costs

As shown in [6], a high number of HITs has to be issued in a large crowdsourcing task. The following factors may significantly increase the costs:

- Number of baseline HITs to be issued
- Number of additional HITs for quality control (e.g., for performing majority votes)
- Need for more qualified, more skilled workers (who expect better paid HITs) [19]
- Higher task complexity or higher task effort (more laborious tasks need higher payment in order to attract suitable workers) [19]

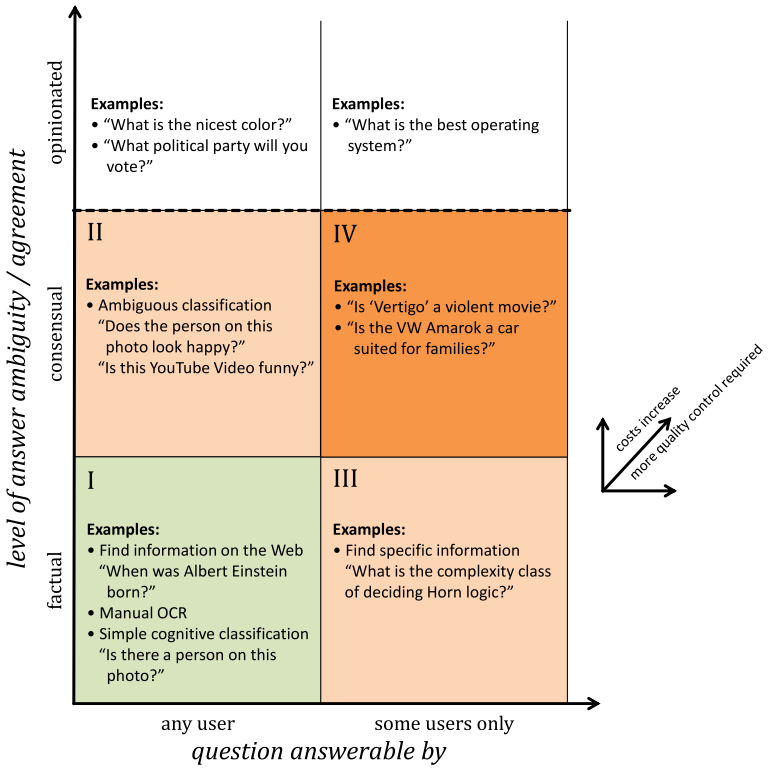

Fig. 1 Four general crowdsourcing scenarios for information extraction classified by restrictions to the potential user groups and level of the ambiguity of "correct" answers, difficulty and expected costs increase along with increasing quadrant number

3.4 Classifying Crowdsourcing Tasks

In this section, we briefly classify crowdsourcing tasks into different scenarios based on two major discriminating factors: (a) the degree of agreement required for deciding if a given crowdsourced task/question is solved/answered correctly and (b) the degree of required restrictions with respect to the group of possible crowd workers. This classification is sketched in Fig. 1.

Using this classification, many popular information extraction tasks can be characterized with respect to their associated style of crowdsourcing. Here are some examples:

- *Named entity detection*. The mere detection of words that directly refer to some named entity reminds of exercises often performed by pupils in elementary school, and thus can be performed easily by human crowdworkers. Moreover, there typically is very little room for ambiguity in results. Therefore, this task is located in quadrant I (= any user, factual).
- *Named entity recognition*. Named entity recognition usually requires understanding the basic structure of a sentence or paragraph. A matching crowdsourcing task could be stated as "What is the Wikipedia page of the entity referred to as ... in the following paragraph: ...". Depending on the ambiguity/clarity (e.g., "IBM" vs. "the computer company with the three letters") and domain (e.g., common newspaper vs. specialized scientific article) of the text to be analyzed, this task could be located in any of the four quadrants. However, tasks solvable by existing algorithms would almost exclusively be located in quadrant I.
- *Speech recognition*. Although research in algorithmic speech recognition has made a lot of progress in recent years, some challenges such as recognizing "untypical" speakers (e.g., dialect, mumbling, or unusually high/low voice) or resolving stylistic devices (e.g., intonation, noises, or made-up words) still remain. Naturally, humans are experts in mastering these challenges in an intuitive way. However, crowdsourcing is more demanding than that as it requires people to make their understanding explicit in detail. For example, a Bavarian dialect that is used as a stylistic device needs to be paraphrased into "(speaks Bavarian)". Similarly, a textual description must be provided for relevant noises such as "(imitates the sound of a saw)". Therefore, advanced speech recognition tasks usually require trained users (quadrants III and IV), while standard tasks (quadrants I and II) often can be handled quite well by existing algorithms.
- *Audio extraction*. Audio extraction typically strives for an explicit description of relevant acoustic features, thus making it very similar to the previous task. Again, the core problem in crowdsourcing is the need for trained users. For example, deriving sheet music from a given audio recording typically requires at least a skilled hobby musician.

3.4.1 *Factual Questions Not Requiring Any Special Skills*

The first scenario is the typical scenario which is addressed by simple crowdsourcing tasks. It is defined by the fact that only very simple tasks not requiring any special skills are to be performed by the workers. Any worker can participate, avoiding the effort of any special recruiting strategies. Therefore, such tasks can be easily facilitated by using generic services such as Amazon Mechanical Turk. Furthermore, the correct solution or answer of a HIT is of what we call "factual" nature. This means that the answer (in practice) is not ambiguous and can easily be evaluated to be either true or false by other humans without disagreement. This criterion has significant impact on the expected result quality and suitable quality control methods. Therefore, it also directly affects the costs, which can be significantly lower in this scenario with comparable result quality. A good example for a task fitting this scenario is manual OCR, e.g., obtaining an e-mail address from a website encoded as an image: Any two honest workers (not making any severe mistakes) will reach the exact same result regardless of their qualification.

Therefore, quality management in this scenario is straightforward:

(a) Assuming that workers are mostly honest, only smaller oversights or mistakes need to be corrected. This can be easily performed by re-issuing HITs and then applying majority votes. The number of questions asked for each majority vote can also be dynamically adjusted such that a minimum threshold of agreement is reached. Therefore, in case of error-prone or even malicious users, a given HIT is re-issued more often (and thus increases the costs).

(b) For filtering malicious users from the worker population, so-called *gold questions* can be used: Gold questions are tasks or questions where the correct answer is known up-front and provided by the task issuer. For each unique worker, some of the gold questions are randomly mixed into the HITs, without informing workers upfront whether a question is gold or not. As soon as the system detects that the number gold questions incorrectly answered by a single worker reaches a certain threshold, the worker is assumed to be malicious and will not receive any payment (and all his/her answers are discarded). As a best practice, a 10 % ratio of gold-to-tasks is generally recommended.[1] Of course, workers must be paid for answering gold questions; therefore using gold questions slightly increases the costs. However, one can safely assume that paying a small overhead for gold questions will pay-off by avoiding the increased number of questions required for compensating malicious users by means of dynamic majority votes.

[1] For more detailed information, see http://crowdflower.com/docs/gold.

Experiment 1: To give some insight into the performance to be expected from such a crowdsourcing task, we conducted an experiment where workers had to find out whether a given movie is a comedy or not [7]. The tasks are issues without restrictions to the general worker pool of Amazon Mechanical Turk. In order to measure the performance of the crowd in the following experiments, as a reference we also obtained expert judgments on genre classification from three major movie community sites (IMDb, Rotten Tomatoes, and Netflix). In total, we obtained a reference classification for 10,562 movies, and used a random subset of 1,000 movies in this experiment. Analyzing the reference classifications clearly showed that movie genres generally are consensual information (instead of factual).

In order to turn this experiment into requiring factual data with no possible disagreement of the correctness of an answer, we required each worker to look up the correct classification on the website of IMDb, the Internet Movie Database. Therefore, this task can be seen as a direct implementation of a simple and straightforward information extraction task realized by using only crowdsourcing. Each HIT was paid with $0.03 and contained 10 movies to be looked up. (Looking up information on websites takes quite some effort; therefore the payment in this experiment is higher than in later experiments. Still, we cannot force users to actually perform the lookup; they may still opt for cheating us and randomly select any option or just guess the answer). Furthermore, we added 10 % gold questions for filtering unreliable workers as described above (i.e., 1,100 movies had to be classified overall). We used the commercial third-party service Crowdflower to handle data quality management (gold questions, majority vote). Therefore, we have no detailed data on how many workers had been automatically excluded without payment, and how many crowd results had been discarded in this process.

We stopped the experiment after the costs reached $30 (10,000 lookups). It turned out that at this time, 93.5 % of all movies had been classified correctly with respect to the original IMDb value, requiring 562 minutes. While this success rate seems to be quite low compared to reliable automated screen-scraping techniques, keep in mind that for the workers of this task it does not matter how the information is encoded on the IMDb website and will result in similar performance even if the genre classification is encoded in an image (e.g., as it is often done with email addresses). Furthermore, even though $30 seems to be much for obtaining just 935 values from a website, paying a programmer for setting up an equivalent automatic IE system may easily become more expensive.

3.4.2 Consensual Questions Not Requiring Any Special Skills

The second scenario is similar to the first one with respect to not requiring any special worker skills. However, now there is significant ambiguity regarding the "correct" answers to questions. Therefore, results generally have to be found by worker consensus as there is no single indisputable correct answer. For example, consider a task where users have to judge if a given YouTube video is funny or not: Obviously, two given users might rightfully disagree on each particular video, but in most cases there will be a community consensus clearly indicating the "correct" result. (Please note that we assume the existence of a clear consensus in this paper, which generally holds for most information extraction tasks.)

Designing such crowdsourcing tasks will imply higher costs and effort compared to the tasks in the first scenario. Mainly, this can be attributed to the fact that a higher number of judgments are required for reaching the required threshold for majority votes. Furthermore, using gold questions becomes less effective or even impossible as "correct" answers are rarely known upfront. But even if there are some examples with clear community consensus which could be used for gold questions, users can hardly be punished for having a different opinion on an obviously ambiguous topic (see Experiment 2). Accordingly, when using gold questions in this scenario, a significantly higher threshold must be chosen.

Of course, malicious users are very quick to realize this dilemma, and therefore tend to try to abuse this kind of tasks for some quick income without high risk for detection. Therefore, for ambiguous tasks a different approach has gained popularity: games-with-a-purpose [20, 21]. Here, the idea is to wrap the HITs into a small game where for example two players pairwise try to individually guess the answer of the other player. If both guessed correctly, both will increase their game score. These games eliminate the central problem discussed previously as malicious users have no incentive to participate in this game. And even if they do, they won't be able to increase their score and their judgments will be filtered in most cases. However, the approach opens up a new problem: How to motivate players to play the game at all? Combined with the games' inflexibility (it has to be laboriously adapted to each new crowdsourcing task, also requiring to attract a new player community in most cases), these approaches are not widespread.

3.4.3 Factual Questions Requiring Special Skills

In this third scenario, we encounter a new challenge in another dimension: tasks that are not solvable by everyone, but require some special skill or background knowledge. In this situation, either the worker pool needs to be limited beforehand such that it only contains accordingly qualified workers (which is often not possible or hard to achieve), or the workers have to be filtered on-the-fly. The obvious approach to filtering is to appeal to worker's honesty by providing an overview of the required skillset, and requiring the users to perform a respective self-assessment. Of course, this approach is highly challenged if a large number of workers in the pool are dishonest. A possible alternative is tapping into respective domain communities instead of relying on the general worker pool. For example, in the Social Web, there are interest groups and specific community platforms for nearly any topic that is of interest to a larger number of people. However, these communities can not readily be harnessed for crowdsourcing, and must laboriously be motivated to partake in such a task. An approach for implicitly exploiting these communities is shown in the last section of this paper.

Therefore, often the situation arises that a general worker pool is filtered beforehand to one's best ability, but still contains a lot of workers which cannot solve all given tasks. For these cases, the option to not solve given tasks must be provided to workers (e.g., offering an "I don't know" option) as otherwise workers will just resolve to providing wrong answers. Furthermore, in our experiments it has been shown that there also needs to be payment even for "I don't know" answers. If not, workers will soon start to complain about and rally against the task and its initiator because they had to spend time and effort, but did not receive payment, which in turn would scare away potential new workers. This can quickly result in the task being completely abandoned by the worker pool. Of course, a paid "I don't know" option will also provide a strong incentive for malicious users to start cheating. This effect is shown in experiment 2 in the next section. More extensive experiments with better prefiltering can be found in [7, 22].

3.4.4 Consensual Questions with Special Skills Required

This scenario combines the challenges of both scenario 2 and 3, i.e. an consensual result is required which imposes strains on quality control, and furthermore tasks are non-trivial, requiring workers with some kind of special skills or background knowledge.

Experiment 2: In this experiment, we evaluated a task similar to Experiment 1. But instead of extracting the respective genre classification from the IMDb website, workers are not allowed to check on the Internet (i.e., they are instructed to use their personal opinion). Consequently, workers can only provide a meaningful assessment for those movies they personally know well (i.e., workers must possess some sort of expert knowledge in the movie domain). In order to retain the worker pool, we allowed users

to use an "I don't know the movie" option while still receiving payment. Again, as always when crowdsourcing is involved, we cannot guarantee that workers perform the task as intended. However, we assume that most workers will either provide a good guess (as intended) or just plainly cheat by randomly selecting any option as other ways of circumventing the task description (e.g., looking up the correct answers on the Web) take significantly more effort. Furthermore, keep in mind that genre classifications are subjective, and therefore a consensual result must be reached in this task. This means that in general it is hard to decide if an answer is correct or not, thus gold questions cannot be easily used. In this particular example, you could use gold for movies where no consensus needs to be reached, e.g., "Schindler's List (1993)" is definitely not a comedy and could be suitable for a respective gold question. However, still "I don't know the movie" would be an acceptable answer to this gold question. In order to simulate the general case, we completely abstained from using gold questions.

We paid \$0.02 per HIT, each consisting of 10 movies. After spending \$20 (10,000 movie judgments) and waiting 105 minutes, it turned out that only 59 % of all movies had been classified correctly compared the consensual expert judgments described in Experiment 1—even when considering majority votes. This low score can be attributed to the fact that most workers spent absolutely no effort on this task, and plainly selected the first available option "this movie is a comedy" in 62 % of all cases (about 30 % of all movie are indeed comedies). Only in 14 % of all cases, the third option "I do not know the movie" was selected. As our sample of 1,000 movies contains many very obscure and less known titles, this result seems to be very unrealistic. A quick survey performed on our more movie-savvy students turned out that they did not know roughly 80–90 % of all movies in the test set. Therefore, we conclude that in this case, due to the lack of quality control measures in form of gold questions and the obvious non-factual nature of the task, our workers are highly dishonest and malicious, and openly try to cheat in order to quickly earn some money. Therefore, alternative considerations for controlling quality are urgently needed for this scenario.

Opinionated Questions In addition to the previously introduced scenarios, also truly opinionated questions and tasks can be crowd-sourced (e.g., "What is the visually most appealing color?" where no community consensus can be reached). In these scenarios, filtering malicious users and controlling quality is even more challenging than in the previously presented cases. However, these scenarios do usually not relate to tasks relevant in information extraction, but have strong ties to market research. Resources on this topic can for example be found in [23].

3.5 Worker Population

In the following, we will briefly discuss the typical worker population, which will be encountered in prominent crowdsourcing platforms by outlining two studies [24, 25] performed on Amazon Mechanical Turk (MTurk) in 2009 and 2010, respectively. This will significantly help to understand worker skills and motivation. Furthermore, some ethical problems are discussed based on these results.

In [24], performed in November 2009, it was shown that most MTurk workers originate from just two countries: 56 % from the US, 36 % from India, while all other countries make up 8 %. Interestingly, in the US, 23 % of all workers come from households which earn more than \$70k. The average age of US workers is 35 years, 63 % are female, and 55 % have some kind of university degree. Furthermore, 62 % state that using crowdsourcing does not change their financial situation, while only 14 % claim that they rely on the money. In both [24, 25] it is claimed, the majority of the US MTurk population is made up of "stay-at home moms who want to supplement the household income; office workers turking during their coffee breaks; college students making a few dollars while playing a game; and recession-hit Turkers doing what they can to make ends meet." This claim is further researched in [25], performed in 2010: 70 % percent of the workers claim that MTurk is a good alternative for watching TV, 40 % state that they do the tasks just for fun, 32 % do them for killing time, while 13 % stated that MTurk is their primary source of income (multiple answers possible).

In contrast, 64 % of the Indian worker population lives in households earning less than \$10k, and are mostly young, well-educated males: 26 years old in average, 66 % male, and 66 % have a university degree. Here, 41 % claim that participating in MTurk does not significantly change their financial situation, while 27 % rely on the money. The Indian worker's motivations are slightly different: 59 % percent claim that MTurk is a good alternative for watching TV (70 % in the US), 20 % claim that they do the tasks just for fun (40 % US), 4 % do it for killing time (32 % US), while 26 % stated that MTurk is their primary source of income (13 % US).

While both studies show that the majority of the 2010 worker population does not rely on the money earned by crowdsourcing but are mostly motivated by different means, both studies also project that in the next years the population will likely shift to including a higher percentage of workers from underdeveloped countries which are depending on the money earned in crowdsourcing. Here, the danger of exploiting the existential struggles of low-income workers for the sake of cheap labor becomes a dominant problem. In order to oppose this development, several platforms

have arisen which specifically aim at providing fair microtask labor in developing countries, aiming at securing a reliable long term income for participating workers while at the same time, ensuring higher trust and quality for the clients (for slightly higher and controlled prices). A popular example for this promising approach is Samasource,[2] which mostly acts as a non-profit crowd-sourcing platform for the African area, focusing on building mirco-labor workspaces and running social programs related to crowdsourcing work. Workers are employed by Samasource, guaranteeing adequate living wages, and turning crowdsourcing into a big chance for otherwise unemployed workers.

4 Hybrid Extraction with Perceptual Spaces

In this section, we will briefly present a hybrid approach combining information extraction, machine learning, and crowdsourcing in order to efficiently and cheaply obtain a large number of complex attribute values. The full details of this approach are published in [7].

The basic challenge is as follows: Given a large set of consumable experience products (e.g., movies, music, books, games, restaurants, or hotels), obtain consensual values for some given perceptual characteristic (e.g., genre classifications, the degree of suspense, whether a song has a positive mood or the suitability for families and smaller children). In this scenario, we assume that the required values are not explicitly available, and therefore cannot be extracted easily by direct means. Therefore, we need to rely either on complex information extraction techniques which heavily employ heuristics and training, or choose to elicit the information directly from the crowd. Using the classification introduced in the previous section, this task can be considered as being quite hard for straightforward crowdsourcing approaches: Traditional quality assurance using gold questions is not feasible due to the consensual nature of required information, and furthermore some specific knowledge is required for estimating the values (e.g., a worker must know a movie to provide a meaningful genre classification; finding suitable workers for more obscure movies will be very challenging). Therefore, we opt for a hybrid approach where crowdsourcing is used for training the underlying information extraction algorithms in an ad-hoc fashion. Hence, we only need rigid quality assurance for the much smaller training set, and also allowing us to use trusted workers or highly qualified (but expensive) experts, while still being able to obtain values for a large number of items without prohibitively high costs.

In the following, we will showcase our approach with the already introduced example of obtaining genre classifications for movies. By relying on information that has been extracted from user feedback on the Social Web, we are able to provide both a clear visual representation of how experience products are perceived by users and also enable intuitive, interactive feedback mechanisms firing the imagination of users. The key idea fueling our approach are perceptual spaces, which are a compressed formal representation of the perceived characteristics of all items in our database. These perceptual spaces are created combining information extraction and recommender system techniques as described in the following section. With perceptual spaces we can implement a system allowing for meaningful semantic exploration. Perceptual spaces can be built from different kinds of (mostly implicit) user feedback publicly available on the Social Web, e.g., ratings, reviews, and link recommendations.

Based on the perceptual space, we then train a machine learning algorithm. Assuming a database query involving a yet-unknown perceptual attribute (e.g., in the movies domain: humor, suspense, or imaginativeness), we first have to understand what this new attribute means. This is best implemented by providing a training sample; i.e., for a small set of movies, the correct judgment of the desired attribute is provided by human experts. This task can be directly crowdsourced, allowing for a quick and efficient ad-hoc adaption of the extraction process (e.g., to obtain a different perceptual attribute). However, ensuring high quality of the training sample is very important for the effectiveness of the learning algorithm. Furthermore, this crowdsourcing task is anchored in quadrant IV of the classification space of the previous section, already requiring experts and a user consensus. As a result, only trusted and competent workers (i.e., workers who have proven their honesty and knowledge, and are therefore more expensive per HIT) should be used, with multiple redundant HITs for a subsequent majority vote phase. Therefore, the resulting costs are comparably high for obtaining just a small, but reliable sample.

From here on, the remainder of the extraction process can again be performed automatically: The training sample can be used to train a classifier, which in turn allows us approximate the missing values of all other movies. For the general case of extracting numeric judgments from a perceptual space, we suggest to use Support Vector (Regression) Machines [26, 27], which are a highly flexible technique to perform non-linear regression and classification, and also have been proven to be effective when dealing with perceptual data [28]. After training this machine learning algorithm with our crowdsourced training sample, based on the perceptual space the algorithm establishes a non-linear regression function. This regression function will finally provide all missing data required for the schema expansion. A graphical overview of the whole workflow is given in Fig. 2.

[2] http://samasource.org/.

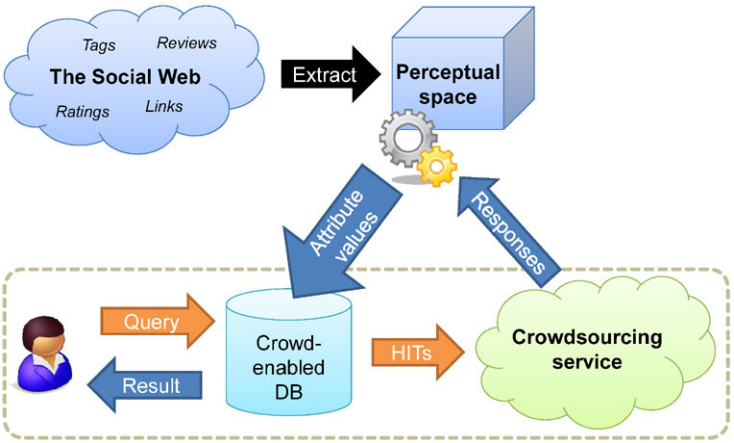

Fig. 2 Architecture of our hybrid approach

4.1 The Social Web as Data Source

The Social Web is on a steep rise. Originally developed for simple communication between people sharing a similar taste, Social Web platforms have become a major innovator of Web technology. With new services being established continuously, and many older ones growing in popularity and reputation, a significant shift in user behavior has occurred: People got accustomed to an active and contributive usage of the Web. Many users now feel the need to express themselves and to connect with friendly or like-minded peers. As a result, general social networking sites like Facebook could amass over 800 million users,[3] while, at the same time, countless special-interest sites developed for music, movies, art, games, or anything that is of interest to any larger group of people. But the real revolution lies in the way people interact with these sites: Following their social nature, millions of people discuss, rate, tag, or vote content and items they encounter on the Web or in their daily lives. Therefore, "I Like" buttons, star scales, or comment boxes have become omnipresent on today's Web pages.

Therefore, the Social Web can be seen as a huge collection of unstructured perceptual data provided by millions of people, created in an implicitly crowd-sourced fashion. In contrast to explicit product descriptions (and respective data models) that could have been created manually by experts or by means of direct crowdsourcing [6], generating data in the Social Web follows different rules: People in the Social Web are entirely intrinsically motivated, and contribute voluntarily as it pleases them. This especially means that this "work" is performed without any explicit compensation or payment. For example, a user finding an interesting online news article might vote for that article on his preferred social site, while a user leaving the cinema after a particular bad movie experience may log onto a movie database, rating the movie lowly, and venting his disappointment in a short comment or review. The motivation for these actions is often anchored in the need for entertainment (i.e., users just spending time in Web, browsing, and commenting for leisure), communication (discussing with peers; expressing one's opinion), and maintaining social contacts (building and maintaining communities with like-minded people).

Therefore, the biggest hindrance in directly using the Social Web as a reliable source of data is that user contributions can neither be controlled nor do they follow a strict schema or guideline. Thus, with respect to processing this data automatically, most of this vast wealth of valuable information just lies dormant. In particular, when dealing with experience products, unlocking this treasure of data would be highly beneficial.

4.2 The Semantics of Perceptual Spaces

Perceptual spaces are built on the basic assumption that each user within the Social Web has certain personal interests, likes, and dislikes, which steer and influence his/her rating behavior [29, 30]. For example, with respect to movies, a given user might have a bias towards furious action scenes; therefore, he/she will see movies featuring good action in a slightly more positive light than the average user who doesn't care for action. The sum of all these likes and dislikes will lead to the user's overall perception of that movie, and will ultimately determine how much he enjoyed the movie and therefore, will also determine how he rates it on some social movie site. Moreover, the rating will share this bias with other action movies in a systematic way. Therefore, one can claim that a perceptual space captures the "essence" of all user feedback, and represents the shared as well as individual views of all users. A similar reasoning is also successfully used by recommender systems [31, 32].

Now, the challenge of perceptual spaces is to reverse this process: For each item being rated, commented, or discussed by a large number of users, we approximate the actual characteristics (i.e., the systematic bias) which led to each user's

[3] http://www.facebook.com/press/info.php?statistics.

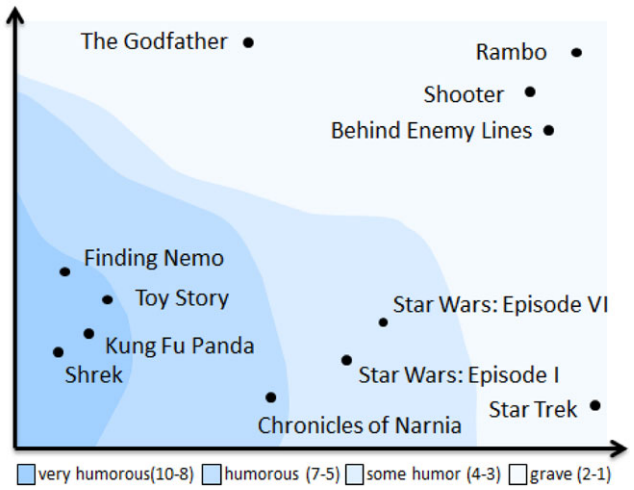

Fig. 3 Example perceptual space. A simplified perceptual space in $\mathbb{R}^2$. While the dimensions do not convey any semantics directly, a judgment of a movie's humor can be extracted

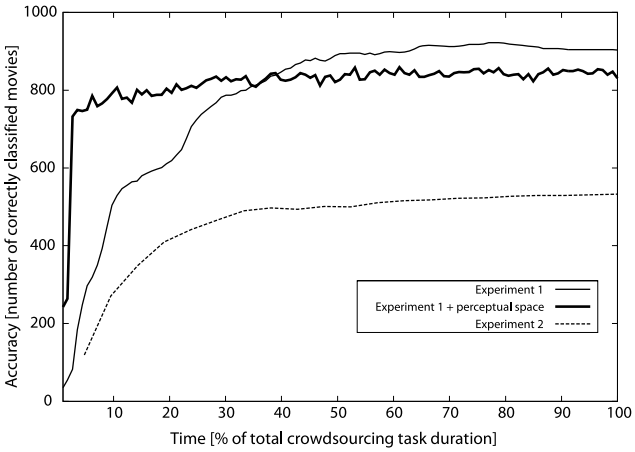

Fig. 4 Quality vs. time results of experiments 1 (562 min), 2 (105 min), and perceptual space

opinion. Formally, we implement this challenge by assuming that a perceptional space is a d-dimensional coordinate space satisfying the following constraints: Each user and each item is represented as a point in this space. The coordinates of a user represent his personality, i.e., the degree by which he likes or dislikes certain characteristics. The coordinates of items, in contrast, represent the profile of that item with respect to same characteristics. Items which are perceived similar in some aspect have somewhat similar coordinates, and items which are perceived dissimilar have dissimilar coordinates (for a simplified example, see Fig. 3).

Next, we assume that a user's overall perception of an item is anti-proportional to the distance of the user and item coordinates, i.e., the "best movie of all times" from a given user's perspective has the same coordinates as the user himself/herself. Of course, a user's likes and dislikes may be slightly unstable due to moods; but on average, this assumption is good enough.

All these ideas can be formalized as an optimization problem, where the variables to be determined are the user and product coordinates, and the criterion to be optimized the degree of fit between user–item distances and observed ratings. By using gradient descent-based methods, this problem can be solved efficiently even on large data sets [33].

4.3 System Evaluation

In this section, we briefly present some evaluations of the costs and the performance of our approach in contrast to using direct crowdsourcing. Mainly, we aim at providing an overview of what to expect from crowdsourcing, and what can be gained by additionally relying on advanced extraction techniques such as perceptual spaces. More thorough evaluations can be found in [7]. We expand on the experiments already introduced in the previous sections.

In Figs. 4 and 5, the results (i.e., percentage of correctly classified movies) of the previously introduced experiments 1 (factual information, no special skills required, lookup of genre in IMDb) and 2 (consensual information, background knowledge required, subjective judgment of genre) are again presented with respect to time required and money spent. Furthermore, we also present the results of our approach based on perceptual spaces. Here, we used a similar setting to Experiment 1 (looking up the genres in IMDb) to create a small high-quality training sample (which increases in size with time and money spent). All remaining movies genres are judged automatically. This experiment representatively simulates scenarios in which we are able to produce a small, but high quality training set. In case that no gold questions and look-ups are possible (e.g., we want to crowdsource some values which are simply not available, but rely on a consensus), this can be achieved by using expensive and trusted domain experts with high thresholds for majority votes. As we can see in the following, even small high-quality training sets allow for a good overall extraction quality, thus such approaches are indeed feasible financially and quality-wise. In the following figures, we can clearly see that by using this approach, we can quickly achieve a high extraction quality, i.e., after just spending $0.32 and waiting 15 minutes, we can already classify 73 % of the 1,000 movies correctly. This figure is tremendously better than when using only consensual judgments by a random worker selection as in Experiment 2. But still, on the long run our result quality is slightly worse when being compared to Experiment 1 (i.e., compared to just looking up the correct genre in IMDb). However, keep in mind that this technique is designed to be used when traditional information extraction fails because the "correct" information is simply not available in the Web. For example when the degree of action on a scale from 1 to 10 in each movie is required, or the family friendliness of each movie is requested, the re-

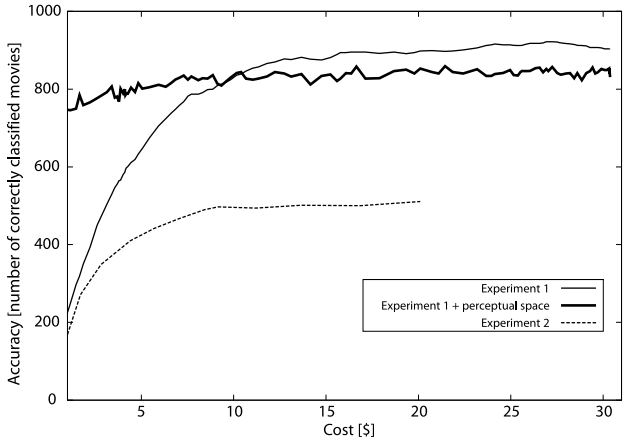

Fig. 5 Quality vs. costs results of experiments 1, 2, and perceptual space

quired information cannot simply be looked up or automatically extracted, but is still implicitly encoded in our perceptual space. In such use cases, our technique is vastly superior to alternative approaches trying to obtain this information.

5 Summary

In this paper, we presented crowdsourcing as a companion technique to traditional information extraction methods. By providing a classification of possible crowdsourcing tasks alongside some exemplary experiments, we pointed out the challenges that need to be overcome in order to reliably use crowdsourcing in such scenarios. In a nutshell, we have shown that crowdsourcing indeed has the potential of being a high-quality complement as long as the tasks are carefully designed with respect to quality control and selection of worker population.

However, the real key to successfully employing crowdsourcing for information extraction lies in true hybrid approaches that transparently blend algorithmic efficiency and finesse with carefully targeted human intelligence. These approaches can easily overcome the limitations of both individual techniques.

References

1. Weikum G, Theobald M (2010) From information to knowledge: harvesting entities and relationships from web sources. In: ACM SIGMOD symp on principles of database systems (PODS), Indianapolis, USA, pp 65–76
2. Chang C-H, Kayed M, Girgis MR, Shaalan KF (2006) A survey of web information extraction systems. IEEE Trans Knowl Data Eng 18:1411–1428
3. Amer-Yahia S, Doan A, Kleinberg JM, Koudas N, Franklin MJ, (2010) Crowds, clouds, and algorithms: exploring the human side of "big data" applications. In: Proceedings of the ACM SIGMOD international conference on management of data (SIGMOD), pp 1259–1260
4. Surowiecki J (2004) The wisdom of crowds. Doubleday, Anchor
5. Doan A, Ramakrishnan R, Halevy AY (2011) Crowdsourcing systems on the world-wide web. Commun ACM 54:86–96
6. Franklin M, Kossmann D, Kraska T, Ramesh S, Xin R (2011) CrowdDB: answering queries with crowdsourcing. In: ACM SIGMOD int conf on management of data, Athens, Greece
7. Selke J, Lofi C, Balke W-T (2012) Pushing the boundaries of crowd-enabled databases with query-driven schema expansion. In: 38th int conf on very large data bases (VLDB). PVLDB 5(2), Istanbul, Turkey, pp 538–549
8. Goodchild M, Glennon JA (2010) Crowdsourcing geographic information for disaster response: a research frontier. Int J Digit Earth 3:231
9. Marcus A, Wu E, Karger DR, Madden S, Miller RC (2011) Crowdsourced databases: query processing with people. In: Conf on innovative data systems research (CIDR). Asilomar, California, USA
10. Etzioni O, Banko M, Soderland S, Weld DS (2008) Open information extraction from the Web. Commun ACM 51:68–74
11. Getoor L, Taskar B (2007) Introduction to statistical relational learning. MIT Press, Cambridge
12. Suchanek FM, Kasneci G, Weikum G (2008) YAGO: a large ontology from Wikipedia and WordNet. J Web Semant 6:203–217
13. Wu F, Weld DS (2008) Automatically refining the Wikipedia infobox ontology. In: Proceedings of the international conference on the world wide web (WWW), pp 635–644
14. Chai X, Gao BJ, Shen W, Doan AH, Bohannon P, Zhu X (2008) Building community Wikipedias: a machine-human partnership approach. In: Int conf on data engineering (ICDE), Cancun, Mexico
15. DeRose P, Shen W, Chen F, Lee Y, Burdick D, Doan AH, Ramakrishnan R (2007) DBLife: a community information management platform for the database research community In: Conf on innovative data systems research (CIDR) Asilomar, California, USA
16. Chai X, Vuong B-q, Doan A, Naughton JF (2009) Efficiently incorporating user feedback into information extraction and integration programs. In: SIGMOD int conf on management of data, Providence, Rhode Island, USA
17. Raykar VC, Yu S, Zhao LH, Valadez GH, Florin C, Bogoni L, Moy L (2010) Learning from crowds. J Mach Learn Res 99:1297–1322
18. Ipeirotis PG (2010) Analyzing the amazon mechanical turk marketplace. XRDS: Crossroads 17:16–21
19. Mason W, Watts DJ (2010) Financial incentives and the performance of crowds. ACM SIGKDD Explor Newsl 11:100–108
20. von Ahn L (2006) Games with a purpose. Computer 39:92–94
21. von Ahn L, Dabbish L (2004) Labeling images with a computer game. In: SIGCHI conf on human factors in computing systems (CHI), Vienna, Austria
22. Paolacci G, Chandler J, Ipeirotis PG (2010) Running experiments on amazon mechanical turk. Judgm Decis Mak 5:411–419
23. Kittur A, Chi EH, Suh B (2008) Crowdsourcing user studies with mechanical turk. In: SIGCHI conf on human factors in computing systems
24. Ross J, Irani L, Silberman MS, Zaldivar A, Tomlinson B (2010) Who are the crowdworkers? Shifting demographics in mechanical turk. In: Int conf on extended abstracts on human factors in computing systems (CHI EA), Atlanta, USA
25. Ipeirotis PG (2010) Demographics of mechanical turk. NYU stern school of business research paper series
26. Drucker H, Burges CJC, Kaufman L, Smola A, Vapnik V (1997) Support vector regression machines. Adv Neural Inf Process Syst 54:155–161
27. Smola AJ, Schölkopf B (2004) A tutorial on support vector regression. Stat Comput 14:199–222
28. Jäkel F, Schölkopf B, Wichmann FA (2009) Does cognitive science need kernels? Trends Cogn Sci 13:381–388

29. Keeney RL, Raiffa H (1993) Decisions with multiple objectives: preferences and value tradeoffs. Cambridge University Press, Cambridge
30. Kahneman D, Tversky A (1982) The psychology of preferences. Sci Am 246:160–173
31. Hofmann T (2004) Latent semantic models for collaborative filtering. ACM Trans Inf Syst 22:89–115
32. Koren Y, Bell R (2011) Advances in collaborative filtering. Recommender Systems Handbook, 145–186
33. Gemulla R, Haas PJ, Nijkamp E, Sismanis Y (2011) Large-scale matrix factorization with distributed stochastic gradient descent. In: ACM SIGKDD int conf on knowledge discovery and data mining (KDD), San Diego, USA. Technical report RJ10481, IBM Almaden Research Center, San Jose, CA, 2011. Available at www.almaden.ibm.com/cs/people/peterh/dsgdTechRep.pdf