Master Thesis

Crowdsourced Product Descriptions and Price Estimations

Steve Aschwanden
Dammstrasse 4
CH-2540 Grenchen
steve.aschwanden@students.unibe.ch
05-480-686

Supervisor
Dr. Gianluca Demartini
C312, Bd de Pérolles 90
CH-1700 Fribourg
demartini@exascale.info

Grenchen, April 7, 2014

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all the principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Steve Aschwanden, 05-480-686

Grenchen; April 7, 2014: ____________________________

(Signature)

Acknowledgements

I like to acknowledge ...

Abstract

I like to acknowledge ...

Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

eBay Inc.[1] is one of the world's largest online marketplaces and reported 128 million active users worldwide during the last quarter of the year 2013. Online auction platforms make consumer-to-consumer transactions possible. The seller can present articles by uploading pictures and describing them. The creation of an auction is time consuming and needs a lot of investigations. Searching for descriptions on the internet or finding a selling price for the same or similar article, for example. In 2005, Jeff Howe and Mark Robinson created a term called 'Crowdsourcing' which is a combination of the words crowd and outsourcing. The idea behind the term is to outsource different tasks, which are difficult to solve by machines, to the crowd. To reduce the costs of collecting information for an article to sell on an auction platform, tasks will be created and outsourced to the crowd. Amazon Mechanical Turk[2], short MTurk, is a crowdsourcing marketplace which enables requesters to publish human intelligence tasks (HITs). The workers can solve these tasks and earn money for good work.

1.1 Statement of the problem

The first step of creating an online auction is mostly to take pictures of the corresponding item. This help the buyers to get information about the state and quality of the article. After that the item needs a short and clear description, some properties (category, state) and a starting offer. If the seller wants to create a lot of different auctions, the whole procedure is time consuming and boring. A price estimation of an article can be difficult, because the background knowledge is missing and other auctions to compare aren't available at any time. Machines aren't able to solve all these steps by them self, because the spectrum of the articles is huge and image processing methods aren't capable to classify them all correctly. To get all the needed parts of an online auction, a human powered approach is necessary. Crowdsourcing platforms provide the possibility to solve tasks, which are difficult to handle for a computer.

1.2 Existing research

tbd

[1] http://www.ebay.com

[2] http://www.mturk.com

1.3 Goals and objectives

The thesis has the following goals and their corresponding objectives:

- **Collect auction item properties by the crowd**
    - Analyse the composition of an auction item on eBay and select the parts which can be crowdsourced
    - Form a ground truth including different auctions created by real online auction platform users by using the eBay API
    - Study literature which covers similar crowdsourcing problems
    - Design and publish tasks on Amazon Mechanical Turk to gather data from the crowd
    - Evaluate the quality of the generated content
- **Try to improve the initial solution by implementing a hybrid approach**
    - Search for image processing or machine learning methods which can simplify and/or support a human intelligence task
    - Implement the methods and adapt the design of the tasks
    - Publish the new tasks on the same crowdsourcing platform
    - Evaluate the results and compare them to the first solution

If the main goals of the thesis are fulfilled, some *optional* goals can be covered by the thesis:

- **Implement a web application which combines the created subtasks to a complete workflow**
    - Find a web application framework which provide an API in the same programming language as the Amazon Mechanical Turk API
    - Create a workflow which put all the subtasks together to an overall solution
    - The user can manage the items (upload pictures to create new items, edit and remove items) and directly create an online auction

1.4 Organization

The thesis is splited into several chapters:

- eBay online marketplace
- Crowdsourcing
- Evaluation
- Conclusion

Chapter 2

eBay online marketplace

2.1 History

eBay was founded 1995 in San Jose (CA) as AuctionWeb by Pierre Omidyar. One year later, eBay bought a third-party licence from Electronic Travel Auction to sell plane tickets and other travelling stuff. During the year 1996, over 200'000 auctions were available on the website. At the beginning of 1997 the number of auctions exploded (about 2 million articles). In the same year the company get their well-known name eBay and received 6.7 million dollar from the venture capital firm Benchmark Capital. The company went public on the stock exchange on September 21, 1998 and the share price increased from 18 to 53.5 dollar on the first day of trading. Four years later the growth continues and eBay bought the online money transfer service PayPal. eBay expanded worldwide in early 2008, had hundred millions of registered users and 15'000 employees. Today, the firm is one of the world's largest online marketplaces. During the fourth quarter of the year 2013 about 128 million active users were reported. A cell phone was sold every 4 seconds, a pair of shoes every 2 seconds and a Ford Mustang every 55 minutes.

2.2 Auction item composition

Every eBay user has the possibility to create auctions for different kind of items. To present the article, the seller has to provide accurate information about it. The standard eBay auction consists of the following fields:

- **Title** The title of the item is limited to 80 characters. The sellers should use descriptive keywords to clearly and accurately convey what they are selling
- **Description** The description is the opportunity to provide the buyers with more information about the item
- **Category** An item can have multiple predefined categories. eBay provides a list of categories which the seller can select
- **Condition** The condition of the item is dependent on the selected category. eBay provides different condition schemas. For clothing items the seller can select between 'New with tags', 'New without tags', 'New with defects' or 'Pre-owned'. For other categories like books, other condition values are present: 'Brand new', 'Like new', 'Very good', 'Good', 'Acceptable'
- **Pictures** To visualise the item the auction creator can upload up to twelve pictures. The first image is important, because it appears next to the item's title in the search result. The pictures will be stored for 90 days on the eBay servers.
- **Shipping costs** The seller has to tell the future buyers how much shipping will cost. There are three possibilities:
  - Free shipping
  - Flat shipping, same cost to all buyers
  - Shipping rate tables, eBay calculates the cost for every individual buyer dependent on the location
- **Duration** An auction can have a duration of 1, 3, 5, 7 or 10 days. If the item has a fixed price, the auction is finished if a buyer is willing to pay this price.

- **Pricing** The seller can select a starting price and then the bidding will start at this price. A 'But it now' option is also available. The buyer can skip the bidding process.
- **Payment** The seller has to select the desired paying method like 'PayPal' or 'Payment upon pickup'

2.3 APIs

eBay provides multiple APIs for developing third party applications. This allows developers to search for auctions or create listings over the XML format. Three main interfaces are available:

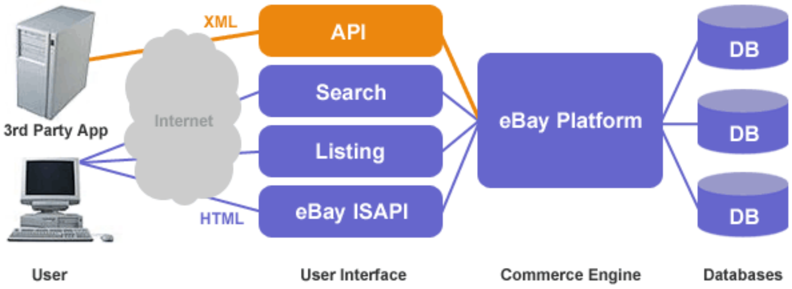


Figure 2.1: eBay API overview

2.3.1 Trading API

Developers use the Trading API to build applications such as selling and post-sales management applications, manage user information, and initiate the item purchase flow on eBay. The API is available in .NET, Java, PHP and Python.

2.3.2 Shopping API

The Shopping API provides a search engine for user information, popular items and reviews. The API is available in PHP and Python. Example calls for this API are:

- *findProducts()*: Search for products by keywords or ProductId
- *GetSingleItem()*: Buyer specific view of an item
- *GetUserProfile()*: Get the user profile and feedback information

2.3.3 Finding API

The Finding API provides access to the next generation search capabilities on the eBay platform. The developer can search and browse for items based on keyword queries, categories or an image. The API is available in .NET, Java and Python. Example calls for this API are:

- *findCompletedItems()*: Find the items which are listened as completed or no longer available on eBay
- *findItemsByCategory()*: Find items in a specific category

- *findItemsByImage()*: Find items which have a high similarity to a given image

2.3.4 Example

The following listing in Python illustrate the functionality of the Finding API. The developer has to register to the eBay developers program first. After that, a application ID can be created. This is necessary to get access to the eBay databases. A functioning Python environment and the additional eBay Python SDK are requirements to successfully execute the example

```
1  from ebaysdk.finding import Connection as Finding
2  from ebaysdk.exception import ConnectionError
3  import json
4
5  try:
6      api = Finding(appid=’Universi-3c25-4b4e-b3e6-8c2568808b12’)
7      api.execute(’findCompletedItems’, {
8          ’keywords’: ’ford mustang’,
9          ’itemFilter’: [
10             {’name’: ’ListingType’,
11              ’value’: ’Auction’},
12             {’name’: ’Currency’,
13              ’value’: ’USD’},
14             {’name’: ’SoldItemsOnly’,
15              ’value’: ’true’},
16         ],
17         ’sortOrder’: ’StartTimeNewest’,
18         })
19     response = json.loads(api.response_json())
20
21     print response[’searchResult’][’item’][0]
22
23 except ConnectionError as e:
24     raise e
```

Listing 2.1: eBay Finding API example

The initialisation of the application is done on line 6. A correct application ID is required. Then the API call *findCompletedItems()* is executed with some keywords and filter options. Only the newest auctions with at least one bidder and a payment in US dollar will be returned. The function *response_json()* (on line 19) returns the first 100 items by default. At the end, the first result will be printed to the console. Here is a shorter simplified version with the most important fields of the output:

Name	**Value**
itemId	281273507096
title	2014 Hot Wheels Super Treasure Hunt 71 Mustang Mach 1
categoryName	Diecast-Modern Manufacture
shippingType	Calculated
currentPrice	18.5 USD
bidCount	1
paymentMethod	PayPal
conditionDisplayName	New
startTime	2014-02-25T04:32:17.000Z
endTime	2014-02-25T05:27:14.000Z

Table 2.1: eBay Finding API example output

Chapter 3

Crowdsourcing

3.1 Introduction

3.2 Platforms

3.2.1 Amazon Mechanical Turk

The project was introduced in 2005 and is part of the Amazon Web Services. Requesters can post tasks known as HITs (Human Intelligence Tasks) which can be solved by workers (Amazon uses another term: Turkers). MTurk provides a web-based user interface and a couple of APIs in different programming languages (.NET, Java, Python, PHP, Perl, Ruby) to manage tasks. The first action of the requester is to create a HIT consisting of mandatory fields:

- **Title** The requester must describe the idea of the HIT in at most 128 characters
- **Description** A more detailed description of the task which cannot be longer than 2'000 characters
- **Question** Every task has to contain questions to collect information from the crowd. The requester can decide between three question data structures
  - QuestionForm The simplest form to create questions in a HIT. MTurk uses a special XML language to define tasks which has some restrictions. For example, JavaScript and CSS are not allowed
  - ExternalQuestion MTurk will display a requester defined external webpage and the answers to the questions will be collected on the external website and send back to MTurk. This question data structure is used to overcome some restrictions of the platform like using JavaScript or to display CSS defined content
  - HTMLQuestion This structure is a mixture between QuestionForm and ExternalQuestion. The requester hasn't to host an external website to provide a HTML based form
- **Reward** If the workers will successfully completing the HIT then they will receive a predefined amount of money from the requester
- **Assignment** duration in seconds The time in which the workers have to complete the task after they have accepted it. The time has to be between 30 seconds and one year
- **Lifetime in seconds** The lifetime of a HIT defines the amount of time a task is acceptable for the workers. After the time elapsed, the HIT will no longer appears in the search results

and some important, optional fields:

- **Keywords** Comma separated keywords which describes the task (max. 2'000 characters)
- **Max assignments** Number of times a HIT can be completed. The default values is one
- **Qualification requirement** Requesters can define requirements to process a task for the workers. Only workers who have more than 100 approved assignments can start working on a requesters HIT for example

After the tasks are designed the requesters have to test them on the Amazon Mechanical Turk Developer Sandbox platform which is a simulated environment. If the requester is happy with the appearance of the HIT, the task can be published on the productive MTurk platform. Turkers have now the possibility to accept the HITs and complete the assignments until the lifetime is expired. After the HIT is completed, the requesters can take a look at the results and have to decide if they want to accept or reject the work. The workers will receive the predefined amount of money only for an accepted task.

3.2.2 Crowdflower

The platform for large-scale data projects was founded in 2007. Crowdflower has over 50 labor channel partners, Amazon Mechanical Turk for example, where the created tasks are published. The partner websites or communities are responsible to manage the registration and payment of their workers. The company offers enterprise solutions and enables a higher degree of quality control. 'Gold standard data' (Quality control - Honey pots) and 'Peer review' are two provided quality control techniques. 'Peer review' gives the requesters the chance to improve the data by a second pass. A workflow management tool helps to link different jobs together. At the time of writing these lines over one billion tasks are completed by workers domiciled in 208 different countries. Also big companies like eBay uses the Crowdflower service for their projects [4]. Over the past years, the company has completed over 15 projects. The improvement of the product categorisation algorithm was one of them.

3.3 Patterns

3.3.1 Find-Fix-Verify

The Find-Fix-Verify pattern was introduced by the Soylent paper [3]. The pattern divide the overall task into three stages. During the Find stage, the workers will identify patches of work done by the crowd or create new patches. For example, the workers has to select a sentence which seems to be incorrect and will need further investigations during the Fix phase. Some workers will revise the identified patches and try to provide some alternatives. The last step of the pattern will present the generated alternatives during the Fix stage to a few new workers in a randomize order. The answer with the most votes (plurality voting) will be used to replace the identified patch during the first phase. The creators of the new suggestions will be suspended so that they can't vote for their own input. To illustrate the meaning of the Find-Fix-Verify pattern, the implementation of Soylent will be discussed (Figure 3.1). The approach begins by splitting a text into paragraphs. During the Find stage, the workers has to identify candidate areas for shortening in each paragraph. If a certain number of workers has selected the same area then this patch goes to the next stage. Every worker in the Fix stage has to present a shorter version of the identified patch if possible. He has also the possibility to say that the text can't be reduced. During the last step, the crowd has to select rewrites which has significant spelling, style or grammar problems or change the meaning of the sentence significantly. At the end they remove these patches by majority voting.

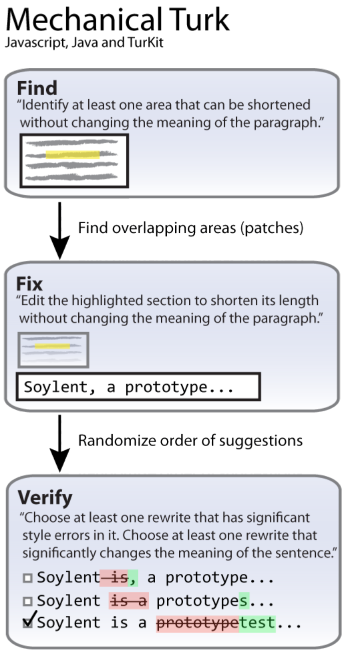

Mechanical Turk
Javascript, Java and TurKit
Find
"Identify at least one area that can be shortened without changing the meaning of the paragraph."
Find overlapping areas (patches)
Fix
"Edit the highlighted section to shorten its length without changing the meaning of the paragraph."
Soylent, a prototype...
Randomize order of suggestions
Verify
"Choose at least one rewrite that has significant style errors in it. Choose at least one rewrite that significantly changes the meaning of the sentence."
□ Soylent ~~is~~, a prototype...
□ Soylent ~~is a~~ prototypes...
☑ Soylent is a ~~prototype~~test...

Figure 3.1: Soylent Fix-Find-Verify pattern

3.3.2 Iterative

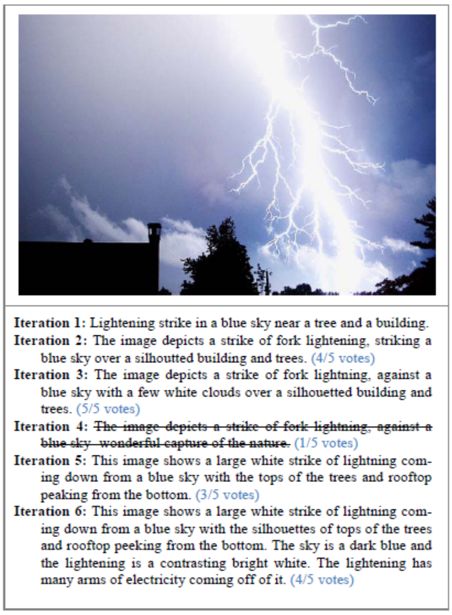
Most of the published assignments on MTurk are independent, parallel tasks. But also iterative, sequential tasks can be useful. The authors of the TurKit paper [10] implemented a tool which make iterative tasks possible. They developed an example application for creating an image description (Figure 3.2). During the first iteration, the worker will contribute the initial description of the provided image. The next iteration will show the initial description and a request to improve it. A few workers will evaluate the extension of the description by voting. If the extended description doesn't receive enough votes then the iteration will be ignored. The final description is generated after a fixed number of iterations. To make the iterative solution possible, the crash-an-rerun programming model was introduced by the authors of the paper. This model allows a script to be re-executed after a crash without generating costly side-effects. That means, if there is a crash during the second iteration of an iterative problem the first iteration will be skipped after re-running the script. TurKit is able to persist the state of the program and will never repeat the successfully completed task. This is helpful for prototyping algorithms.

3.4 Design

If requesters want to create new HITs then they have to consider some design guidelines [1, 2]:

- **Be as specific as possible in the instructions** If the requesters ask the workers 'Is a Ford Mustang a sports car?' is not the same as they ask 'Can a Ford Mustang accelerate from 0 to 100 km/h in 3 seconds or less?', because the second one is clearer and more precise. Sometimes it is useful to hire a technical writer for phrasing task instructions

- **Instructions have to be easy to read** Instructions should be split into multiple subtasks



Iteration 1: Lightening strike in a blue sky near a tree and a building.

Iteration 2: The image depicts a strike of fork lightening, striking a blue sky over a silhoutted building and trees. (4/5 votes)

Iteration 3: The image depicts a strike of fork lightning, against a blue sky with a few white clouds over a silhouetted building and trees. (5/5 votes)

Iteration 4: ~~The image depicts a strike of fork lightning, against a blue sky wonderful capture of the nature.~~ (1/5 votes)

Iteration 5: This image shows a large white strike of lightning coming down from a blue sky with the tops of the trees and rooftop peaking from the bottom. (3/5 votes)

Iteration 6: This image shows a large white strike of lightning coming down from a blue sky with the silhouettes of tops of the trees and rooftop peeking from the bottom. The sky is a dark blue and the lightening is a contrasting bright white. The lightening has many arms of electricity coming off of it. (4/5 votes)

Figure 3.2: Iterative image description created by TurKit

and presented as a bulleted list entries

- **Provide examples** The best way to present the idea of a task is to show one or multiple examples. This can help to avoid uncertainties, for example if instructions are misinterpreted or the workers have wrong expectations

- **Mention what won't be accepted** If a worker should write a paragraph about an encyclopaedia article, the requester can allude in the instructions that copying contents from other website are prohibited

- **Tell the workers which tools they should use**

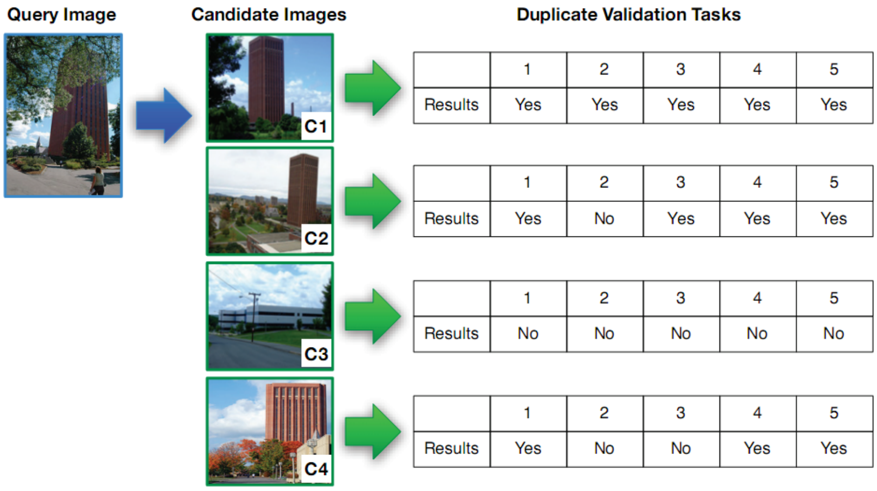
- **Give the workers the possibility to write down a feedback about the task** This is important to improve the design of the tasks or can help to detect spammers

- **Iterative and incremental development** of tasks The first draft of a task will never be perfect. With the feedbacks and results of the previous iterations, the next one will contain improvements which should avoid foregoing mistakes or design failures

MTurk best practices, Iteration, Very important, Instructions are the key

3.5 Hybrid

A lot of information systems use a hybrid crowdsourcing technology. The combination of human intelligence and machine algorithms can lead to powerful information systems which can't be realised by a pure machine approach. In most cases, the crowd is responsible to process the created content of machine algorithms or generate input data for them. A closer look at the CrowdSearch [14] project helps to illustrate the idea of hybrid systems. The developers implemented an image search system for cell phones. First, the system uses an automated image search to generate a



	1	2	3	4	5
Results	Yes	Yes	Yes	Yes	Yes

	1	2	3	4	5
Results	Yes	No	Yes	Yes	Yes

	1	2	3	4	5
Results	No	No	No	No	No

	1	2	3	4	5
Results	Yes	No	No	Yes	Yes

Figure 3.3: CrowdSearch hybrid image search approach

set of candidate pictures. These are packed into multiple identical tasks for validation by humans and published on Amazon Mechanical Turk (Figure 3.3). A simple majority voting is used to eliminating errors. After the validation of the results, the resulting image will be presented to the user. The drawbacks of such systems are that the hybrid approach generates additional costs for involving humans and the delay between publishing the tasks and receiving the corresponding results. The users of CrowdSearch can define a deadline before they query an image and the system will always return a result after the time is expired, independent if the crowd sourced tasks are completed or not.

3.6 Quality control

Determine the quality of completed tasks by the crowd is a very important. Workers can be lazy or are spammers which want to earn the money for free or a minimal amount of work. To evaluate the performance of a single worker, several techniques are available.

3.6.1 Majority voting

To reduce the errors of single workers, majority voting can be used. If a majority has the same answer to a question, the requester can assume that the answer is correct. To break tie situations, a expert is necessary.

3.6.2 Honey pots

The requesters include trap questions where they know the correct answer. If the answer of a single worker is incorrect, the requester can exclude the results or reject the task. But it's not always possible to generate honey pots.

3.6.3 Qualification test

MTurk provides the possibility to include a qualification test at the beginning of tasks. The worker has to pass the test to have access to the real tasks and the corresponding rewards. The results of the test can be compared to an answer key automatically or by the requesters themselves. The additional effort and the determent of some workers are drawbacks of this procedure.

3.7 Workflow

A workflow is a set of tasks which are interconnected and easier to solve by crowd. The output of a single subtask will be used for one or multiple subsequent subtask. The output of the last element of the flow is the result of the entire complex task. It exists a lot of literature which covers the problematic of finding and interconnect subtask:

The process of decomposing complex tasks into simpler ones is not always easy and need a lot of clarifications. The developers of the Turkomatic [9] tool had an obvious idea and source the workflow decomposition out to the crowd. The workers should decide how the final workflow should look like and what are the belonging tasks. The system consists of two major parts. The meta-workflow is used to design and execute workflows by applying the price-divide-solve (PDS) procedure. The workers has to recursively divide the complex task into smaller ones until they are simple enough. After this step the workers will solve the generated tasks and other workers are asked to check the solutions. At the end, the results are combined into a cohesive answer. The second part of the Turkomatic system allows a visualisation of the created workflows and an edit function to manually adapt the crowdsourced results.

Another idea pursues the developers of CrowdForge [8]. They designed a framework to create a workflow by using several partition, map and reduce steps. The partition step split a larger task into smaller subtasks, the map step let one or more workers process a specified task. The results of the workers are merged into a single output during the reduce step. For example, the workers should write an encyclopaedia article about a given topic (Figure 3.4). The authors of the paper solved this problem by the presented partition/map/reduce steps. First, the partition step asks the workers to create an outline of the article by defining section headings (e.g. "History", "Geography"). During the map phase, multiple workers are asked to provide a single fact about the section (e.g. "The Empire State Building celebrated its 75th Anniversary on May 1, 2006" if it's an encyclopaedia article about "New York" and the section heading is "Attractions"). The workers has to piece the collected facts together to a completed paragraph during the reduction step.
The CrowdForge prototype is written in Python using the Django web framework and Boto, an interface to the Amazon Web Services which is available in Python. The user can define complex flows by creating HIT templates (which can be either a partition, map or reduce task) and dependencies between the templates to define a flow. Flows are implemented as Python classes. The prototype is also responsible for the sequential coordination between the HITs (including data transfer). Multiple independent flows can be executed simultaneously. One of the limitations is that CrowdForge does not support iteration or recursion. The further development of the project was suspended in 2011.

The same crew developed CrowdWeaver [7] which is an advancement of the CrowdForge project. They use CrowdFlower, an other crowdsourcing platform, instead of Amazon Mechanical Turk. On CrowdFlower, the requester can create tasks on multiple markets (including MTurk). Flows can

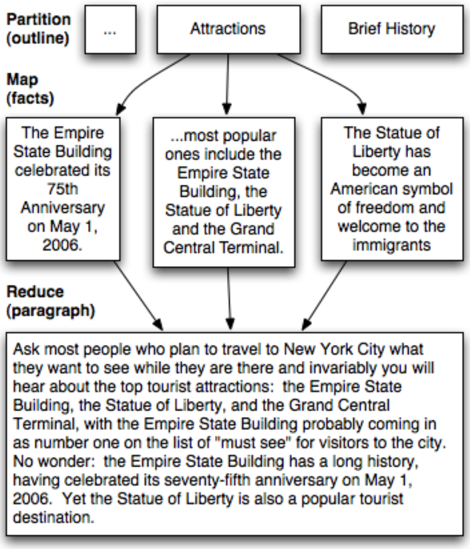

Partition (outline)
...
Attractions
Brief History
Map (facts)
The Empire State Building celebrated its 75th Anniversary on May 1, 2006.
...most popular ones include the Empire State Building, the Statue of Liberty and the Grand Central Terminal.
The Statue of Liberty has become an American symbol of freedom and welcome to the immigrants
Reduce (paragraph)
Ask most people who plan to travel to New York City what they want to see while they are there and invariably you will hear about the top tourist attractions: the Empire State Building, the Statue of Liberty, and the Grand Central Terminal, with the Empire State Building probably coming in as number one on the list of "must see" for visitors to the city. No wonder: the Empire State Building has a long history, having celebrated its seventy-fifth anniversary on May 1, 2006. Yet the Statue of Liberty is also a popular tourist destination.

Figure 3.4: CrowdForge example workflow

be created visually and doesn't assume any programming skills. Another feature is the tracking and notification of crowd factors, for example latency or price.

Jabberwocky

3.8 Incentives

3.8.1 Gamification

The ESP game [12] makes the labelling of any kind of images on the web possible. There are no guidelines to provide images and no computer vision method exists which can handle the diversity of all images. Search engines are dependent on accurate image descriptions to represent relevant results. Therefore, another approach was introduced by the article. A online, web-based game was developed to attract workers. Two players are randomly assigned to label the same image simultaneously. There is no possibility to communicate with the game partner. Now, every player has to guess the description of the image independently without using the 'Taboo words'. These words are evaluated by a prior round and will be ignored for the actual turn. If there is a match between both players, the score will be increased and another image description was detected. The discovered word will only be taken as a valid description and 'Taboo word' if a predefined number of players had the same agreement. The duration of the whole game is 150 seconds and both parties can guess as many images as possible within this time. During a period of four months, the game was played by 13'630 people and 1'271'451 labels for 293'760 images were generated. These numbers show the power of the idea. The players (crowd) didn't know what's going on behind the scenes and they also didn't realise the purpose of their inputs.

3.8.2 Socialisation

"Social factors such as the desire to feel a sense of involvement and 'belong' to a social group, and the forming and maintaining of interpersonal bounds, are a fundamental human need. Empirical studies also show that social motivation is an important driver for people taking part in online activities, ranging from knowledge contribution to providing emotional support." [15]

One example project which use this social incentive is 'stackoverflow' [1]. People are able to post questions about computer programming issues and other users will provide their help for free. Good answers will receive votes from other contributors and the person who asked the question is authorised to mark an answer as accepted. Hard workers can earn reputation points from other users for questions, answers or edits. A higher reputation score will unlock advanced functionalities. Another way to earn respect from other users is to gather badges. That are achievements which are available in three levels: bronze, silver and gold. "Answer score of 100 and more", "Asked a question with 10'000 views" or "Visited the site each day for 30 consecutive days" are example activities which will be rewarded with badges. The two presented rewards motivate the users of the website to contribute as much content as possible. The community itself is controlling the quality of the answers, because experts can remove wrong or low quality statements. Normal users can penalise improper answers by not voting for them. The service sorts answers based on the votes in descending order and the worse evidences will be ignored by the customers.

3.8.3 Unintended by-product

Data from the crowd is collected as an involuntary by-product of the mainly purpose. One of the most famous projects is reCAPTCHA [13] which is further development of the well known Captcha[2] idea. The method will show distorted characters, which can't be recognised by the OCR (Optical Character Recognition) software, to the internet users. The reCAPTCHA acts like a normal Captcha, but the inputs will be used additionally to improve text recognition systems. Another project from the same inventor is Duolingo[3]. Luis von Ahn has the vision to translate every page in the web into every major language. He hides the main purpose of the service behind a free foreign language learning program. Companies remunerate the founder of the project for translated documents.

3.8.4 Financial reward

Another possibility to attract workers is the good, old money. Crowdsourcing platforms offer to pay them for accepted tasks. If the payment is too low then workers won't process the tasks. High rewards will attract spammers who deliver bad quality work to collect as much cash as possible in a short amount of time. A research paper from Yahoo [11] investigates the relationship between financial reward and the performance of the crowd. They found out that a higher payment increases the quantity of the work and not it's quality. They proposed to use other incentives like enjoyable tasks or social rewards if possible, because the quality of work is the same or better than financial driven approaches. A second advice is that requesters should use as less money as possible if only a payment of the workers is possible. Based on the fact that work will be done faster but not better if a higher gratification will be paid. Amazon itself doesn't provide numbers, but suggest to take a look at similar HITs to compare the reward [2]. Also a good strategy is to proof how long it takes to complete the own tasks and calculate then how many tasks can be done in one hour.

[1] http://stackoverflow.com/

[2] http://www.captcha.net/

[3] https://www.duolingo.com

Different analyses [5, 6] show that the median wage is \$1.38/hour and the average wage \$4.8/hour. The Mechanical Turk Tracker website[4] was developed by the author of one of these statistics [6] and it's possible to calculate the average cost per HIT for a specific day. At March 10th 2014, the website tracked 236'370 completed HITs with a total reward of 23'110\$ and the average of \$0.097/HIT. These numbers should help the requesters to find an initial price for their tasks. But there is no general formula to calculate the right costs for an HIT. If the initial price is too low, the workers will ignore these tasks and try to find others with a better revenue/expense ratio. This results to higher completion times. In this case the requesters should increase the reward. On the other side, if the tasks will be completed very fast and the results are not like expected then a decrease of the reward can be helpful.

3.9 Demography

The workers on the MTurk platform will be hidden behind an identification number, no details about gender or country of residence are available. To get detailed information about the workers, researchers from the University of California published surveys in the form of HITs [?] and presented their results in 2010. They observed the crowd about 20 months and detected some changes over time. The number of Indian workers raised significantly within one year and approximately one third of the workers are from there. The majority of the turkers is located in the United States (56Unfortunately, the presented facts are four years old, but no current numbers are available for the Amazon Mechanical Turk platform.

[4] http://mturk-tracker.com

Chapter 4

Implementation

4.1 Pure approach

4.1.1 Ground truth

Real eBay auctions were collected by the API to generate the ground truth for the crowdsourcing experiments. The online auction platform divides their items in eight main categories: Motors, Fashion, Electronics, Collectibles & Arts, Home & Garden, Sporting Goods, Toys & Hobbies and Deals & Gifts. The ground truth consists of seven items from every category with the exception of the Motor's and Deals & Gits sections, because the API can't search for items in these categories. First, some keywords were created to touch the desired category: "Swiss Watch" (Fashion), "Smartphone" (Electronics), "Football Trading Card" (Collectibles), "Coffee Machine" (Home), "Soccer shoes" (Sporting Goods), "Action Figure" (Toys) and "Handbag" (Fashion). The goal was also to have gender specific and neutral items. An action figure is used by male persons normally, the handbag by females and a smartphone by both. The Finding eBay API provides the method *findCompletedItems* which takes keywords as a parameter and returns a list of completed auction items. The Python script search for the first sold item which uses US dollar as currency, has a description longer than one-hundred characters and contains of at least three images. Only three images were kept, because most of the auctions present the items with a top, front and side view. Another reason is the clarity for the crowdsourcing tasks. Every ground truth entry has the attributes title, description, category, condition, price and image one to three. The following table represents the final ground truth for the further experiments:

ID	Image 1	Image 2	Image 3
1			
	Title	NIB 45 EURO $$ PUMA SPORT WRISTWATCH SWISS WATCH MOVEMENT LOVE+FOOTBALL	

	Description	ITEM IS BRAND NEW, FROM THE FIFA WORLD (MUNDIAL) FOOTBALL GAMES. WATCH HAS THREE INTERCHANGEABLE TOP COVER , EACH ONE REPRESENTING THE FLAG OF A TEAM PLAYING AT THE FIFA WORLD (MUNDIAL) FOOTBALL GAME PLUS ONE BLACK COVER IF YOU DON'T WISH TO WEAR THE FLAG COLORS INCLUDED IN THE PACKAGE AS SHOWN IN MY PICTURES. ITEM IS BRAND NEW NEVER USED WITH ORIGINAL BOX PAPERS AND INSTRUCTION ON HOW TO USE THIS WATCH.GREAT GIFT IDEA OR GREAT WATCH FOR THE SPORT LOVERS.ITEM COMES WITH WARRANTY FOR 90 DAYS FROM US AND MANUFACTURER WARRANTY OF 2 YEARS IS INCLUDED IN THE BOX.INSTRUCTIONS ARE IN DUTCH,ENGLISH,FRENCH,ITALIAN, CZECH,GERMAN, PORTUGUESE,SPANISH,HUNGARIAN,CHINESE AND JAPANESE.
	Category	Jewelry & Watches:Watches:Wristwatches
	Condition	New with tags
	Price	4.99

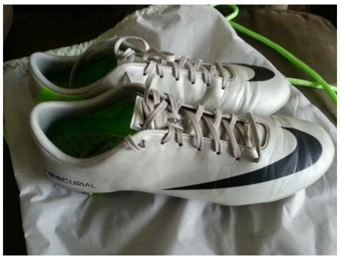
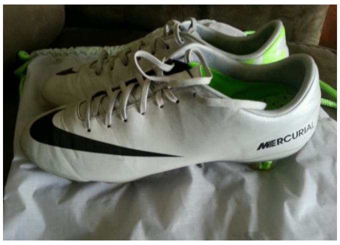
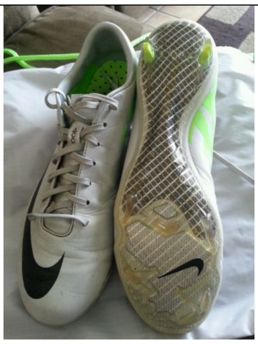
ID	Image 1	Image 2	Image 3
2			
	Title	Apple iPhone 4 - 16GB - Black (Unlocked) Smartphone	
	Description	16GB Black iPhone 4, unlocked by carrier. This was an AT&T phone so it is GSM, can be used internationally. This phone was manufacturer refurbished and then only used for about a week, so it is basically in perfect condition. Includes original packaging, 30-pin USB connector and charger.	
	Category	Cell Phones & Accessories:Cell Phones & Smartphones	
	Condition	Used	
	Price	185.0	

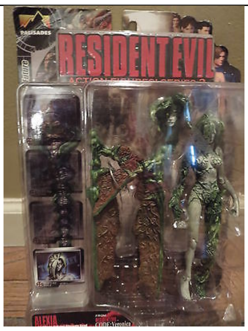
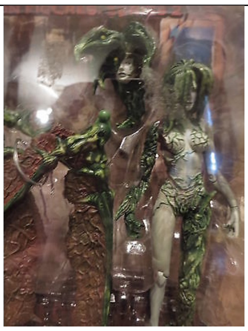
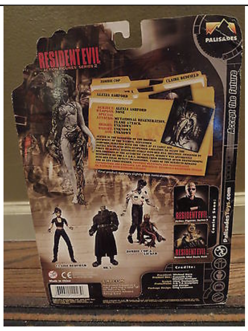
ID	Image 1	Image 2	Image 3

3

	Title	Lot of (13) 2013 Bowman Sterling Autograph Auto Relic Jersey Games Used
	Description	This is for a 2013 Bowman Sterling Lot of 13 Game Used Relics and Autos. You get the exact cards that you see in the pictures. PLEASE PAY BY PAYPAL WITHIN 24 HOURS OF AUCTIONS END OR ITEM WILL BE RELISTED. S+H IS 3.99 WITH DELIVERY CONFIRMATION PLEASE CHECK OUT MY OTHER AUCTIONS
	Category	Sports Mem, Cards & Fan Shop:Cards:Football
	Condition	Brand New
	Price	27.0

ID	Image 1	Image 2	Image 3
4			
	Title	Nespresso Aeroccino Plus & Citiz Coffee Machine Red	
	Description	Nespeesso Aeroccino Plus & Citiz Coffee Machine Fully automatic brewing and milk frothing in two sleek, compact units. Works exclusively with Nespresso's premium coffee capsules, which are easy to order for delivery within two business days (for details, visit www.nespresso.com). Innovative Thermoblock technology with stainless-steel heating element guarantees precise temperature control. A 19-bar pressure pump ensures maximum extraction of flavor. Adjustable tray accommodates cups of various sizes (from small mug to travel cup). Removable water tank for easy refilling. Energy-save mode gradually reduces power if unit is left on. Includes Aeroccino Plus milk frother, which quickly heats milk for consistently perfect foam. Frother has two whisk attachments and an auto shutoff feature. Espresso maker: ABS plastic housing. 14 1/2" x 5" x 11" high. 34-fl.-oz.-cap. water tank. 10 lb. 1200W. Milk frother: Stainless-steel and plastic construction. 4" diam., 6-3/4" high. 8-oz. cap. 550W. This product is intended for use in the United States and Canada and is built to United States electrical standards. Posted with eBay Mobile	
	Category	Home & Garden:Kitchen, Dining & Bar:Small Kitchen Appliances:Coffee & Tea Makers:Espresso Machines	
	Condition	New	
	Price	201.0	

ID	Image 1	Image 2	Image 3
5			
	Title	Nike Mercurial Vapor IX FG - Soccer Shoes Cleats - Metallic Platinum	
	Description	This is a pair of used Nike Vapor IX. They come with the string bag. In overall good condition with some signs of use. Clean and no smells. Mens size 7.5. Shipping is $10.00 and includes tracking. I accept PayPal for payment.	
	Category	Sporting Goods:Team Sports:Soccer:Clothing, Shoes & Accessories:Shoes & Cleats:Men	
	Condition	Pre-owned	
	Price	76.99	

ID	Image 1	Image 2	Image 3
6			
	Title	RARE Series 2 Palisades Resident Evil Code Veronica Alexia Action Figure	
	Description	This RARE and HARD TO FIND action figure will make and AWESOME collectable for any Resident Evil fan! This specific figure is part of the Resident Evil Code Veronica series. Alexia comes complete with Wings, Tail and Alternate Head to Transform into Alexia III and Logo Base. Great item for any RE fan!!! This item is still in its original packaging, unopened and unused. There is very slight wear around the cardboard edging from years of storage and a little adhesive residue on the plastic, most likely from a price sticker. Overall this item is in excellent condition!	
	Category	Toys & Hobbies:Action Figures:TV, Movie & Video Games	
	Condition	New	
	Price	90.0	

ID	Image 1	Image 2	Image 3
7			
	Title	Black Coach purse leather GUC serial number H050-9247	
	Description	Pre-owned Black Coach hobo purse. GUC just because I did use it a couple of times. No stains,marks, or tears. Great condition!!!	
	Category	Clothing, Shoes & Accessories:Women's Handbags & Bags:Handbags & Purses	
	Condition	Pre-owned	
	Price	35.0	

Table 4.1: Ground truth for pure crowdsourcing

4.1.2 Tasks workflow

The macro task of generating an auction on eBay is splited into four simpler micro task:

- Generate a title for the auction item based on the provided images of the item
- Generate the description of the item
- Find the category of auction item
- Define a starting price for the auction

[Create image of workflow]

4.1.3 Task design

4.1.3.1 Title

4.1.3.2 Description

4.1.3.3 Category

4.1.3.4 Price estimation

Another idea to estimate the starting price is inspired by a German TV game show. The candidate has to predict the cost of an article. After the first guess, the game master answers with 'higher' or 'lower' until the right guess occur or the time is running out. If the player finds out the correct price then she/he will win the object. The idea of the show is modified to implement a game with a purpose, similar to the ESP game project [12]. The general procedure of the game is the following:

1. The system waits until two independent players are connected and ready to play
2. A few pictures, title and description of the article are displayed and the players had to read them first

3. Then the game starts and a first guess of the price will be shown by the system
4. Both users have to decide if the real price is higher or lower than the displayed one
5. Dependent on the previous response, the system will present a higher or lower price until the countdown is expired or there are no guesses left
6. The players will receive a score dependent on the difference of the price estimation. A smaller difference leads to a higher score, a higher one to a lower score

The first guess of the system will be the mean value μ of a large number of sold items on eBay. The value can be determined by the eBay API. The guessing structure will be implemented as a directed binary tree. The root node represents the mean value and every following child node will have a lower (left child) v_l or higher (right child) value v_r, determined by the value of the parent node v_p and the depth d of the tree. The following formula calculates the values of the nodes:

$$v_l(v_p, d) = v_p - \frac{\mu}{2^d} \tag{4.1}$$

$$v_r(v_p, d) = v_p + \frac{\mu}{2^d} \tag{4.2}$$

The leafs are integer values which can't be divided by two and represents the final guess of a player. If the time is up and the guesser doesn't reach a leaf node, the value of the actual node is taken. The score of the price prediction is determined by a scoring function s, where x_1 and x_2 are the price estimations of player 1 and 2.

$$s(x_1, x_2) = 1 - |\varphi(x_1) - \varphi(x_2)| \tag{4.3}$$

The function φ is responsible to normalise the estimations (interval from 0 to 1).

$$\varphi(x) = \frac{x}{2\mu} \tag{4.4}$$

The function is also used to weight the different estimations for the same product. If n rounds are played for a given object, the final price t is calculated:

$$t = \frac{1}{\sum_{k=1}^{n} s(x_{k1}, x_{k2})} \left(\sum_{i=1}^{n} s(x_{i1}, x_{i2}) \frac{x_{i1} + x_{i2}}{2} \right) \tag{4.5}$$

The reliability r of the price estimation is the mean score of all played games for the same object:

$$r = \frac{1}{n} \left(\sum_{i=1}^{n} s(x_{i1}, x_{i2}) \right) \tag{4.6}$$

4.2 Hybrid approach

4.2.1 Ground truth

4.2.2 Tasks workflow

4.2.3 Task design

4.2.3.1 Title

4.2.3.1.1 Finding

Title	Give auction items a title based on several images
Description	Take a look at three pictures of an auction item and write down a predicative title
Keywords	image, description, picture, item, title, auction
Reward	0.05 USD
Lifetime	7 Days
Assignment duration	30 minutes
Assignments	3
Qualification requirement	Location is US

Table 4.2:

Overview

Question 1 The goal of the requester of the HIT is to generate eBay auctions by MTurk workers based on images of the item.
Instructions: Write a title which describes the item(s) on the pictures, inside the empty text field below and keep in mind the following remarks:

- Use descriptive keywords to clearly and accurately convey what you're seeing. You're allowed up to 80 characters, but you don't need to use them all
- Include the item's brand name, artist, or designer if possible
- Include item specifics. For example, include size, color, condition, and model number if possible
- Use correct spelling
- Don't use all caps
- Take a look at the provided example

4.2.3.1.2 Voting

Overview

Title	Vote for an auction item title based on several images
Description	Take a look at three pictures of an auction item and select a title
Keywords	image, description, picture, item, title, vote, auction
Reward	0.02 USD
Lifetime	7 Days
Assignment duration	30 minutes
Assignments	3
Qualification requirement	Location is US

Table 4.3:

Question 1 The goal of the requester of the HIT is to generate eBay auctions by MTurk workers based on images of the item.
Instructions: Select the title which describes clearly and accurately the auction item(s) on the pictures. Keep in mind the following restrictions:

- Do not select a title which contains wrong information
- Do not select a title which has spelling mistakes (e.g. Sny Playstation 4, black)
- Do not select a title which is written in capital letters (e.g. SONY PLAYSTATION 4, BLACK)

If you think that no title is suitable for the item(s) then select None.

Question 2 Give reasons for your selection:

- Explain why the selected title is the best description of the item(s) on the pictures
- Explain the deficits of the non-selected titles (e.g. incorrect information)

Attention: Only HITs with meaningful reasoning will be approved

4.2.3.2 Description

4.2.3.3 Category

4.2.3.4 Price estimation

4.2.4 Pre-processing

4.2.5 Feature extraction

4.2.6 Feature selection

4.2.7 Classifiers

Chapter 5

Evaluation

5.1 Pure approach

5.2 Hybrid approach

Chapter 6

Conclusion

6.1 Improvements

6.2 Future work

Bibliography

[1] Omar Alonso and Matthew Lease. *Crowdsourcing for Information Retrieval: Principles, Methods, and Applications.* In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 1299–1300. ACM, New York, NY, USA (2011). ISBN 978-1-4503-0757-4. doi:10.1145/2009916.2010170. URL `http://doi.acm.org/10.1145/2009916.2010170`.

[2] Amazon. *Requester Best Practices Guide.* URL `http://mturkpublic.s3.amazonaws.com/docs/MTURK_BP.pdf`.

[3] Michael S. Bernstein/ Greg Little/ Robert C. Miller/ Björn Hartmann/ Mark S. Ackerman/ David R. Karger/ David Crowell/ and Katrina Panovich. *Soylent: A Word Processor with a Crowd Inside.* In *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology*, UIST '10, pages 313–322. ACM, New York, NY, USA (2010). ISBN 978-1-4503-0271-5. doi:10.1145/1866029.1866078. URL `http://doi.acm.org/10.1145/1866029.1866078`.

[4] CrowdFlower. *Case study - eBay* (2013). URL `http://cdn2.hubspot.net/hub/346378/file-522132326-pdf/docs/CF-eBay-CS.pdf?t=1392311997000`.

[5] John Joseph Horton and Lydia B. Chilton. *The Labor Economics of Paid Crowdsourcing.* In *Proceedings of the 11th ACM Conference on Electronic Commerce*, EC '10, pages 209–218. ACM, New York, NY, USA (2010). ISBN 978-1-60558-822-3. doi:10.1145/1807342.1807376. URL `http://doi.acm.org/10.1145/1807342.1807376`.

[6] Panagiotis G. Ipeirotis. *Analyzing the Amazon Mechanical Turk Marketplace. XRDS*, 17(2):16–21 (December 2010). ISSN 1528-4972. doi:10.1145/1869086.1869094. URL `http://doi.acm.org/10.1145/1869086.1869094`.

[7] Aniket Kittur/ Susheel Khamkar/ Paul André/ and Robert Kraut. *CrowdWeaver: Visually Managing Complex Crowd Work.* In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, CSCW '12, pages 1033–1036. ACM, New York, NY, USA (2012). ISBN 978-1-4503-1086-4. doi:10.1145/2145204.2145357. URL `http://doi.acm.org/10.1145/2145204.2145357`.

[8] Aniket Kittur/ Boris Smus/ Susheel Khamkar/ and Robert E. Kraut. *CrowdForge: Crowdsourcing Complex Work.* In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 43–52. ACM, New York, NY, USA (2011). ISBN 978-1-4503-0716-1. doi:10.1145/2047196.2047202. URL `http://doi.acm.org/10.1145/2047196.2047202`.

[9] Anand Kulkarni/ Matthew Can/ and Björn Hartmann. *Collaboratively Crowdsourcing Workflows with Turkomatic.* In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, CSCW '12, pages 1003–1012. ACM, New York, NY, USA (2012). ISBN 978-1-4503-1086-4. doi:10.1145/2145204.2145354. URL `http://doi.acm.org/10.1145/2145204.2145354`.

[10] Greg Little/ Lydia B. Chilton/ Max Goldman/ and Robert C. Miller. *TurKit: Human Computation Algorithms on Mechanical Turk.* In *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology*, UIST '10, pages 57–66. ACM, New York, NY, USA (2010). ISBN 978-1-4503-0271-5. doi:10.1145/1866029.1866040. URL `http://doi.acm.org/10.1145/1866029.1866040`.

[11] Winter Mason and Duncan J. Watts. *Financial Incentives and the "Performance of Crowds".* *SIGKDD Explor. Newsl.*, 11(2):100–108 (May 2010). ISSN 1931-0145. doi:10.1145/1809400.1809422. URL `http://doi.acm.org/10.1145/1809400.1809422`.

[12] Luis von Ahn and Laura Dabbish. *Labeling Images with a Computer Game.* In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pages 319–326. ACM, New York, NY, USA (2004). ISBN 1-58113-702-8. doi:10.1145/985692.985733. URL `http://doi.acm.org/10.1145/985692.985733`.

[13] Luis von Ahn/ Benjamin Maurer/ Colin McMillen/ David Abraham/ and Manuel Blum. *reCAPTCHA: Human-Based Character Recognition via Web Security Measures.* *Science*, 321(5895):1465–1468 (2008). doi:10.1126/science.1160379. URL `http://www.sciencemag.org/content/321/5895/1465.abstract`.

[14] Tingxin Yan/ Vikas Kumar/ and Deepak Ganesan. *CrowdSearch: Exploiting Crowds for Accurate Real-time Image Search on Mobile Phones.* In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, pages 77–90. ACM, New York, NY, USA (2010). ISBN 978-1-60558-985-5. doi:10.1145/1814433.1814443. URL `http://doi.acm.org/10.1145/1814433.1814443`.

[15] Lixiu Yu/ Paul André/ Aniket Kittur/ and Robert Kraut. *A Comparison of Social, Learning, and Financial Strategies on Crowd Engagement and Output Quality.* In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work &#38; Social Computing*, CSCW '14, pages 967–978. ACM, New York, NY, USA (2014). ISBN 978-1-4503-2540-0. doi:10.1145/2531602.2531729. URL `http://doi.acm.org/10.1145/2531602.2531729`.

Appendix A

Some Appendix

A.1 README

```
Fuzzily classify twitter messages using storm and store to cassandra
===


Setup Cassandra (on ubuntu):
---
1. Make sure oracle JDK is installed (1.6+): https://help.ubuntu.com/community/Java#Oracle_Java_7
2. Add the DataStax repository key to your aptitude trusted keys.
> $ curl -L http://debian.datastax.com/debian/repo_key | sudo apt-key add -
3. Install Cassandra:
> sudo apt-get update && sudo apt-get install cassandra
4. Create keyspace and tables:
> cqlsh
> run commands from src/main/resources/createDatabase.txt

Build Runnable jar
---
1. Open a terminal window, navigate to pom.xml directory (project root)
2. Execute the following command:
> mvn clean compile assembly:single
3. In target/, a runnable jar tsfc.jar is created

Run Program
---
> java -jar tsfc.jar <<comma separated list of topics to watch (without whitespace)>>
```