CA' FOSCARI UNIVERSITY OF VENICE
Department of Environmental Sciences, Computer Science and Statistics
MSc Program in Computer Science

Thesis

Student: Zohreh Khojasteh Ghamari - 840127

A Crowdsourced System For User Studies In Information Extraction

Thesis Supervisor: Prof. Salvatore Orlando

Academic Year 2012-2013

A Crowdsourced System For User Studies In Information Extraction

Zohreh Khojasteh Ghamari - 840127

March 2014

Abstract

The use of crowdsourcing platforms for evaluation the relevance of search results has become a significant strategy that presents results so quickly with spending trivial amount of money. At first appearance, when we are talking about Crowdsourcing, there will be a big issue and that is the priority of using human intelligence instead of running a script or application. A decent answer is being such kind of jobs that involve some element of interpretation, synthesis, or evaluation and humans perform well on them in contrary with computers performance which is poor. For instance, humans could better describe the action taking place in a photo, or determine whether a word is linked to a correct Wikipedia page considering the meaning of other words in a certain sentence, where this would be missed by a computer. In this thesis, we take a set of tweets, where some subsequence of words or in the other word spots are annotated with possible meaning/entities which are linked with Wikipedia pages. Then we survey a sample of people asking them to decide about the possibly perfect Wikipedia page that must be linked with a definite Word/spot. For this end, we create a crowdsourced system in Crowdflower in order to study user behaviors, evaluate the outcome and discuss the results. In particular, the main reason of using Crowdflower instead of other crowdsourcing systems of which, the most famous one is Amazon Michanical Turk, is that CrowdFlower has a lot more channels where we can publish our tasks compared to AMT which is just a single standalone channel. Crowdflower covers both big micro task sites like Amazon Mechanical Turk and smaller channels like Getpaid, Zoombucks etc. In fact tasks are completed within a few hours even if we would have thousands of them. As input data, we had two files, one was containing 1000 tweets, but only 178 of them have one or more spots; and the other was containing 15623 spots annotated with possible Wikipedia articles. In order to input CrowdFlower, we combined the two files into a single

tsv file, containing 208 units of work. A unit corresponds to a single occurrence of a spot in a tweet, that has more than one spot knowing that 81 are linked to only one Wikipedia page. Hence we had 208 spots in 178 tweets. After building the job in crowdfower, it took 71 minutes to get back the reports from. Analyzing the report, we reach some interesting results that by considering them on next jobs we will reach to better results.

Contents

List of Figures

List of Tables

Introduction

Crowdsourcing user studies are so significant for many aspects of the design process and related techniques from informal surveys to very difficult and sensitive laboratory studies. Crowdsourcing is an effort to obtain needed services, ideas, or content from a large group of people by demanding contributions, especially when this group of people is from an online community, rather than from traditional employees or suppliers. Indeed we can say that "Crowdsourcing is the act of taking a job traditionally performed by a designated agent which is usually an employee and outsourcing it to an undefined, generally large group of people in the form of an open call." Crowdsourcing process usually takes part to subdivide tedious work or to fund-raise startup companies and charities, and can also occur offline. In the classic use of the term, tasks are distributed among an unknown group of solvers in order to have an open call for solutions. Users or in the other word crowd submit solutions to the crowdsourcer. Usually these contributors are compensated by money or by being well-known in a certain field. In fact there is no idea about getting solutions from an ascertained group of workers, so the solutions may be gotten from an amateur worker which is working in his spare time or from some experts. Crowdsourcers are motivated by its benefits. One of the advantages is the ability to gather large numbers of solutions and information at a relatively inexpensive cost. Generally users are attracted to contribute to crowdsourced tasks to have social contact, intellectual stimulation, passing time and earning money at the same time. In general, there are plenty of definition about crowdsourcing in scientific and popular literature area, the best integrated definition of it is the following: "Crowdsourcing is a type of participation online activity in which an individual, an institution, a non-profit organization, or company proposes to a group of individuals of varying knowledge, heterogeneity, and number, via a flexible open call, the voluntary undertaking of a task. The undertaking of the task, of variable complexity and modularity, and in which the crowd should participate bringing their work, money, knowledge and/or experience, always entails mutual benefit. The user will receive the satisfaction of a given type of need, be it economic, social recognition, self-esteem, or the development of individual skills, while the crowdsourcer will obtain and utilize to their advantage that what the user has brought to the venture, whose form will depend on the type of activity undertaken". [1]Many people use the term crowdsourcing broadly to describe many different models, such as crowdfunding, ideation platforms, and prediction markets. Moreover there are many differences be-

tween crowdsourcing and outsourcing, however one of the most important differences of them is the elimination of a single point of failure. In fact crowdsourcing is a one-to-many relationship where work is distributed to multiple contributors, and usually submitting multiple answers of the same unit of work till it leads to an agreement. In other word, the probability of inaccurate work decreases with crowdsourcing decreases. In contrast, outsourcing involves a one-to-one relationship where work is contracted to a third-party. We believe that the most famous and popular channel in order to request a work to be crowdsourced is Crowdflower. CrowdFlower is a crowdsourcing service founded in 2009 by Lukas Biewald and Chris Van Pelt [1]. CrowdFlower has scaled its crowd to such a huge community of online contributors which is the biggest in the world. It is partnering with couple of websites that maintain large online communities. These partners are called "Channels." The contributors access CrowdFlower tasks via offer walls on different Channel websites like Amazon Mechanical Turk. The positives about CrowdFlower are that it won't reject contributors easily except when they are really poor in doing works. Instead it will quality screen them out of certain jobs. Moreover almost all of their HITs are auto-approved after 24 hours, so they pay quickly.

In this thesis we use the definition of crowdsourcing as the act of distributing labor amongst a large group of people via online microtasks. Knowing that Microtasking is "dividing a large project into smaller and well defined tasks". Obviously this work will have a lot of advantages and since we know that microtasks typically require human intelligence, so they are performed online by some persons, often with a small amount of research, in contrary with automated through a computer. The most significant advantage of microtasking is that a large volume of work can be completed by the crowd with minimal skills or training.

Right after introducing Crowdsourcing, the matter of quality gets important, In [2] they investigated three basic paraeters of crowdsourcing experiment design : Pay, effort and worker qualifications. Also they discussed about their influence on the quality of the output, measured by accuracy. They found out that experiment designers need to find the right balance between too low pay that results in sloppy work and too high pay that attracts unethical workers. Also when they copared different groups of workers, they found that more qualified workers produce better quality work, so both pay and qualification leads to significant differences in output quality. With respect to effort, they found that while higher effort induces more spam, it also leads to slightly better quality after spam removal than low effort HITs, though this is not statistically significant. At the end they conclude that

increasing pay, reducing effort, and introducing qualification requirements can all help in reducing undesirable behavior.

The most popular crowdsourcing system that is well known almost by every person either amateur computer applicant or professionals is Amazon Mechanical Turk, mainly because it's the best system for paying sufficient for its workers. Since the use of crowdsourcing platforms like Amazon Mechanical Turk for evaluating the relevance of search results has become so popular by thousands of its advantages, there is still the issue of trust for worker's judgments. Mainly poor judgments by workers can occur when they prefer to answer many questions as quickly as possible in order to earn more money in limited time, in the other hand, some workers can be ethical but misinterprets the designer's intent for task, therefore writing the instruction of a job is quite important. It is mentioned that one approach to ensure quality of worker judgments is to include an initial training period and subsequent sporadic insertion of predefined gold standard data (training data). This training data has dynamic learning opportunity for workers, since each worker has to complete 20 query-result pairs[3]. Each worker has to complete this number of query-result pairs successfully before proceeding to test-set questions. The workers are notified that only upon passing this section they will receive payment. Workers will be informed of their mistakes. After this training period, training data is used as periodic screening questions[4]to provide live feedback when workers err. The feedback explains what the correct answer should be and the reason of it. For every 20 query-result pairs a worker saw, they also were exposed to five training data questions in periodic screening. As a worker answers these training data questions, their accuracy will be calculated and use it as an estimate for the worker's "true" accuracy. In [5] they investigated the utility of a micro-task market for collecting user measurements, and discussed design considerations for developing remote micro user evaluation tasks. Although micro-task markets have great potential for rapidly collecting user measurements at low costs, they found that special care is needed in formulating tasks in order to harness the capabilities of the approach.

In [6] they described a new approach to evaluation called Technique for Evaluating Relevance by Crowdsourcing(TERC) which is a crowdsourcing-based alternative to traditional relevance evaluation, in which many online users, drawn from a large community, each performs a small evaluation task. Consequently they found out that the TERC approach is complementary in Information Retrival researchs and provides a flexible and inexpensive method for large-scale editorial relevance judgments. After that, in [7] they showed a series of experiments on TREC data, evaluate the outcome, and

discuss the results. Their position, supported by these preliminary experimental results, was that crowdsourcing is a viable alternative for relevance assessment. Using TREC data, they have demonstrated that the quality of the raters is as good as the experts. Their experience shows that it is extremely important to carefully design the experiment and collect feedback from turkers.

In any way, there is still no established methods to measure the quality of the collected relevance assessments, In [8], they disscussed the components that could be used to devise such measures, such as several sources of evidence that could be used to derive a trust weight for the judgments: topic familiarity and familiarity with the content being assessed, dwell time and changes in the patterns of dwell time, agreement between judges, and the presence and length of comments.

In [9]they talked about their experience using both Amazon Mechanical Turk and CrowdFlower to collect simple named entity annotations for Twitter status updates. As we know Twitter is an informal and abbreviated form in usage of named entity experiment. They developed separate tasks on CrowdFlower and MTurk using a common collection of Twitter statuses and asked workers to perform the same annotation task in order to fully understand the features that each provides, and to determine the total amount of work necessary to produce a result on each service and the found out that MTurk has the advantage of using standard HTML and Javascript instead of CrowdFlower's CML. However MTurk has inferior data verification, in that the service only provides a threshold on worker agreement as a form of quality control. In the other hands, CrowdFlower works across multiple services and does verification against gold standard data, and can get more judgements to improve quality in cases where it's necessary.

In[10] they explore the design and execution of relevance judgments using Amazon Mechanical Turk as crowdsourcing platform, introducing a methodology for crowdsourcing relevance assessments and the results of a series of experiments using TREC 8 with a fixed budget. Their findings indicate that workers are as good as TREC experts, even providing detailed feedback for certain query-document pairs. They also explore the importance of document design and presentation when performing relevance assessment tasks.

In[11]they investigate the design and implementation of effective crowdsourcing tasks in the context of book search evaluation. They observe the impact of aspects of the Human Intelligence Task (HIT) design on the quality of relevance labels provided by the crowd.

As all other systems which in first collaboration, we have to have an account, for working through crowdflower as a task provider , first step is

creating an account too. Afterwards for building a new job, in each job, we have to do three steps: Design job, manage quality, get the results.

Part I
Design job

he first step is Designing the job, in this step we need to do three subsections which is explaining in the following:

1 Data

for uploading data in crowdflower there are two options:

1- Upload file and add source data via a spreadsheet in the format of .csv, .tsv, . xlsx, .ods

2- Pull data and add source data via a data feed in the format of RSS, Atom, XML, JSON.

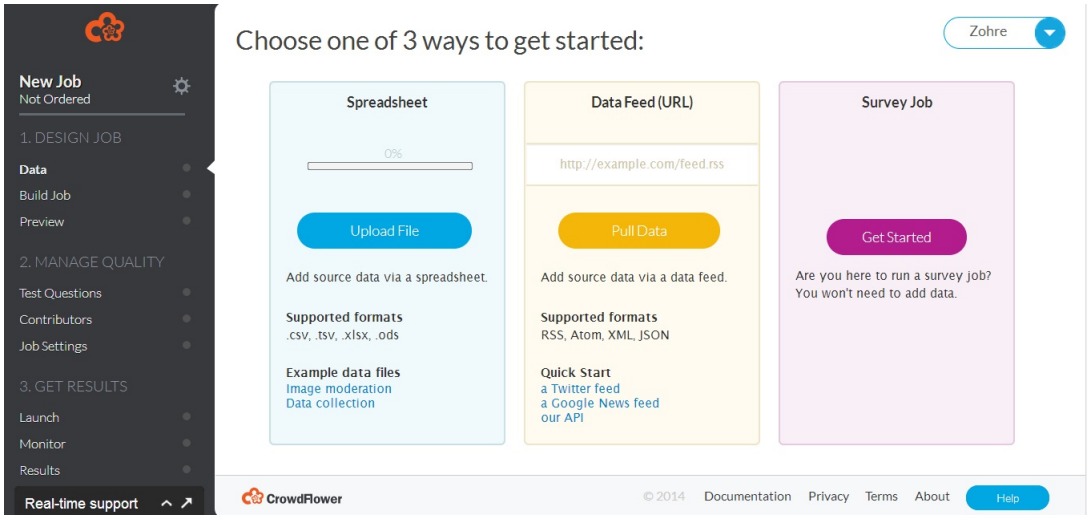


Figure 1: Data

In our work, the input data was made from combination of two files, one was containing 1000 tweets, but only 178 of them have one or more spots; and the other was containing 15623 spots annotated with possible Wikipedia

articles. The combination was a tsv file, containing 208 units of work, ready to upload.

Here is the code we did in Python:

```
from pprint import pprint

def unicode_csv_reader(utf8_data, dialect=csv.excel, **kwargs):
    csv_reader = csv.reader(utf8_data, dialect=dialect, **kwargs)
    for row in csv_reader:
        yield [unicode(cell, 'utf-8') for cell in row]


job_units = csv.writer(open('job-units-5.tsv',"wb"), dialect="excel-tab")
job_units.writerow(['tweet', 'spot', 'start', 'end', 'wiki'])

entities_hash = {}
with open("spot-to-all-entities.tsv","r") as spot_to_entities:
    csvreader = unicode_csv_reader(spot_to_entities, dialect="excel-tab")
    for row in csvreader:
        entities_hash[row[0]] = row[1]

with open("tagged-tweets.json","r") as tagged_tweets:
    for line in tagged_tweets:
        tweet = json.loads(line)
        entities = []
        if not 'entities' in tweet:
            continue
        for spot in tweet['entities']:
            if (spot['spot'] in entities_hash):
                entity = {}
                entity['wiki'] = json.loads(entities_hash[spot['spot']])
                tweet_text = tweet['text']
                job_units.writerow([tweet_text, spot['spot'],
                    spot['start'], spot['end'], json.dumps(entity)])
```

2 Build Job

Second subsection in Designing job is Building the job. Once you've settled on a workflow for your job, it's time to create a form. There are two interfaces for editing a form: The Graphical Editor and the CML Editor.

• The Graphical Editor is the ideal tool for creating simple forms.

• The CML Editor allows you to use code to implement logic and contingencies, HTML, Javascript, and CSS in your forms. In our work, we used this option and created our job by CML Editor.

2.1 Graphical Editor

Without typing any code the Graphical Editor allows you to create a job - Title, Instructions, Insert Your Data, and generate Form Elements There are three different components of the Graphical Editor:

1. Add Title and Instructions
2. Insert Your Data
3. Add Questions

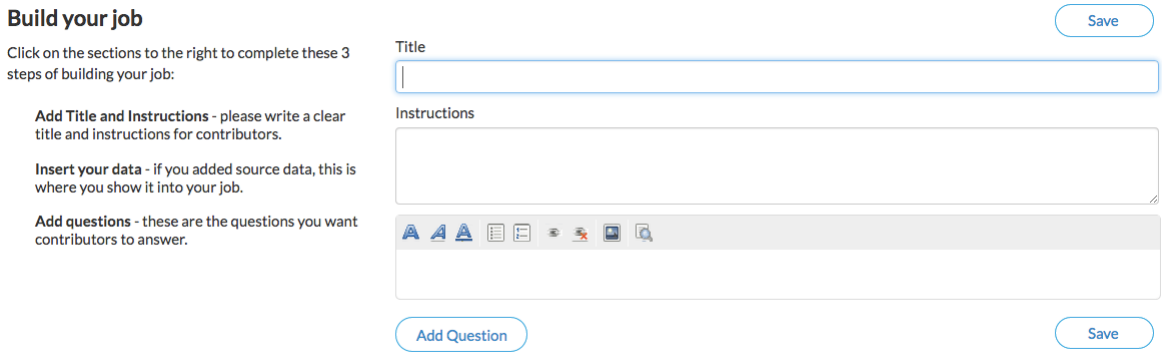


Figure 2: Graphical Editor

The graphical editor has two parts - the right side of the page is used to create the form and the left side of the page shows instructions about each part of the editor. The instructions menu will change in response to which part of the form editor is selected on the right side of the page. After writing title and instructions of your job. Insert Your Data If you added source data, this is where you show it into your job. Click the menu box underneath Instructions to select which data you would like to show in your job. Insert Your Data box A menu will display when you select the Insert Your Data field on the right side of the page. Once selected, all column headers from your data will be shown – add them to your job by selecting them. The selected column headers will display the data values of that column in your form. The text area also allows you to insert links, pictures, and format your text. Add Questions Select the Add Question button at

the bottom of the page to add Form Elements to your task. This tool allows you to use of an array of different fields, such as radio buttons (multiple choice), text labels, text areas, checkboxes, drop down menus, and a rating tool. Test Questions serve the dual purpose of training contributors and monitoring contributor performance. Contributors are given a Trust Score that reflects their accuracy on Test Questions in a given job. This score is reduced when a contributor submits a judgment that does not agree with the answer in a gold unit. Feedback, including the correct answer and an explanation, is provided to the contributor for each incorrect response on a gold unit.

2.2 CML Editor

Use the CML Editor to build custom jobs with HTML, CSS, Javascript, and Crowdflower's handy markup language for forms, CML.

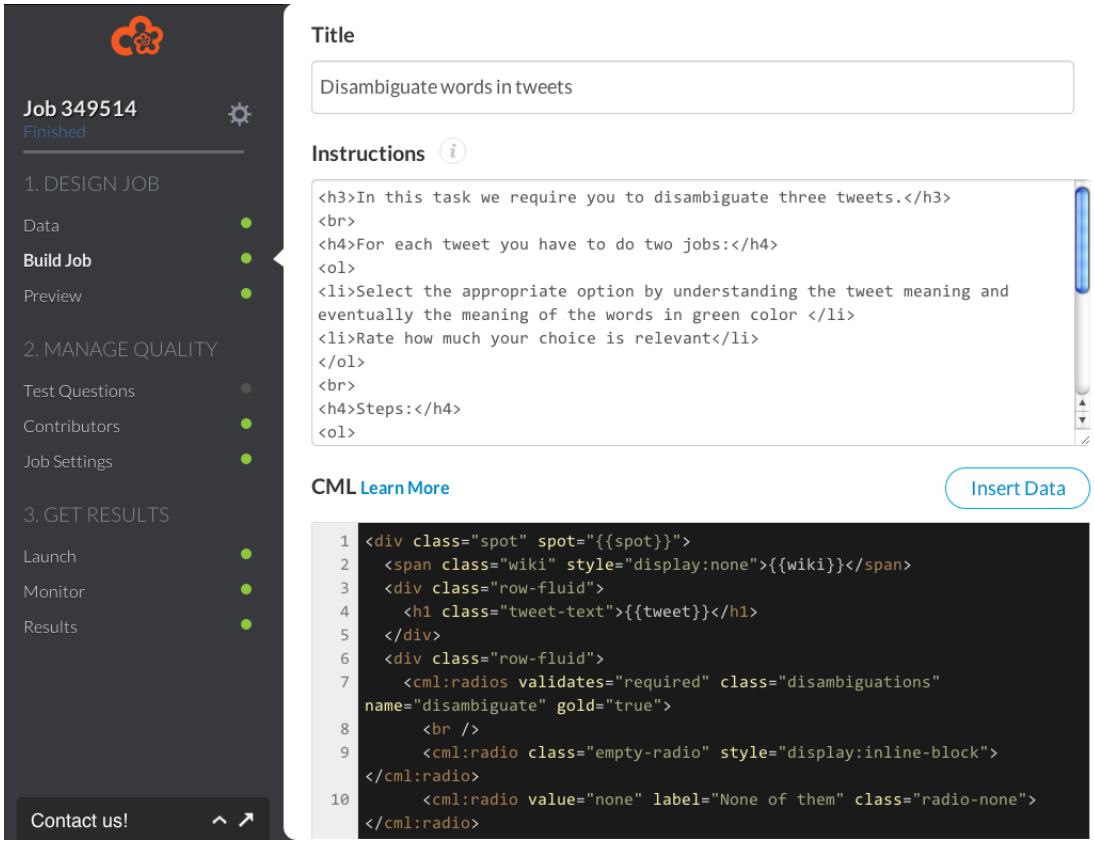


Figure 3: CML Editor

CML is CrowdFlower Markup Language. CML is made up of a set of helper tags, which makes defining forms to collect information from our labor pools quick and painless. The interactive form builder automatically generates most of these helper tags. If you need more control over your forms, or you simply prefer interacting with CrowdFlower through the API, CML is for you. Why CML? CML has 4 main advantages over raw HTML: 1. CML automatically namespaces form elements. Because we display multiple forms in a single page, all form elements must be properly namespaced. CML takes care of that for you. 2. CML lets you write less markup. There is no need to wrap your form elements in containers or add labels, CML writes all the extra markup for you. 3. CML stores meta information specific to the CrowdFlower platform. Gold specification and directives for how you want your data aggregated are specified directly on the form elements. 4. CML makes input validation simple. For instance, add validates="required

numeric" to any CML tag, and you can be sure you'll get only numbers back in your form data for that tag.

Basic tags

•cml:text - Single line text input

• cml:textarea - Multi-line text input

• cml:checkbox - Single checkbox

• cml:checkboxes - Multiple related checkboxes

• cml:radios - Multiple radio buttons

• cml:select - Drop-down menu

• cml:ratings - Multiple radios in a single line

As we said, Crowdflower uses CML and Liquid to generate the form needed for each unit of work. Thus the same CML code is used for all units in the dataset. Crowdflower allows us to use JS (w/ jQuery) and CSS to further customize the CML. JS code is run once on page load knowing that the CML is converted to HTML server side. In CML Editor, After writing the title and instructions of our job, we can use the CML field to customize the way contributors submit data to complete an assignment. HTML can also be used in this place too.

Alternatively Liquid is another markup language developed by Shopify which allows to output values into the CML/HTML code from a unit's row in the dataset, by column name (e.g. {{col_name}}). In fact array data (and JSON objects) cannot be interpreted by Liquid, hence they must be tokenized to strings and then parsed with a Liquid filter (i.e. split). Alternatively JSON objects can be loaded with JS (on page load), if the value is written to a form element (as a string) using Liquid.

Our CML codes are shown in the following:

```
<span class="wiki" style="display:none">{{wiki}}</span>
<div class="row-fluid">
    <h1 class="tweet-text">{{tweet}}</h1>
</div>
<div class="row-fluid">
    <cml:radios validates="required"
        class="disambiguations" name="disambiguate" gold="true">
        <br />
        <cml:radio class="empty-radio" style="display:inline-block">
        </cml:radio>
        <cml:radio value="none" label="None of them" class="radio-none"
        </cml:radio>
    </cml:radios>
```

```
        <cml:group only-if="disambiguate:[none]">
            <cml:textarea validates="required" name="reason"
                label="None of them? Please tell us the reason:"></cml:text
        </cml:group>
    </div>
    <cml:group only-if="!disambiguate:[none]">
        <div class="row-fluid">
            <cml:radios validates="required" name="rating" class="ratings">
                <br />
                <cml:radio label="Very relevant" value="very_relevant"></cm
                <br />
                <cml:radio label="Not too much relevant"
                    value="not_too_much_relevant">
                </cml:radio>
                <br />
                <cml:radio label="Relevant" value="relevant"></cml:radio>
            </cml:radios>
        </div>
    </cml:group>
</div> <!-- /.spot -->


var $ = window.jQuery;

function preg_quote( str ) {
        return (str+'').replace(/([\\\.\+\*\?\[\^\]\$\(\)\{\}\=\!\<\>\|\:])
                "\\$1");
}

function highlight( data, search ) {
        return data.replace(
                new RegExp( "(" + preg_quote( search ) + ")" , 'gi' ),
                        "<span class=\"spot-word\">$1</span>" );
}

$(document).ready(function(){
setTimeout(function(){

  $('.spot').each(function(i) {
    var spot_div = $(this),
```

```
    spot = spot_div.attr('spot'),
    wiki = JSON.parse(spot_div.find('.wiki').text()),
    num_wikis = wiki.wiki.length,
    disambiguations = spot_div.find(".disambiguations"),
    ratings = spot_div.find(".ratings"),
    legend = disambiguations.find(".legend");

// highlight spot in tweet
var tweet = spot_div.find("h1.tweet-text");
tweet.html(highlight(tweet.text(), spot));

// fix legends
legend.html("Please disambiguate the word <span class=\"spot-word\">&qu
    + spot + "&quot;</span>");
ratings.find(".legend")
.html("Please rate the word <span class=\"spot-word\">&quot;"
    + spot + "&quot;</span>");

// load radios
var empty_radio_row = disambiguations.find(".empty-radio")
    .closest(".cml_row");

for (var i = num_wikis - 1; i >= 0; i--) {
  var wiki_title = ((wiki.wiki[i]).title).replace(/_/g, " ");

  var curr_radio = empty_radio_row.clone();
  curr_radio.find("input").attr("value", i); // index of wiki as value
  curr_radio.find("label").append(" " + wiki_title);
  curr_radio.removeClass("empty-row");

  // append wikipedia link
  curr_radio.append("<small><a href=\"http://en.wikipedia.org/wiki/"
    + (wiki.wiki[i]).title
    + "\" class=\"wiki-link\""
    + " target=\"_blank\">view in Wikipedia</a></small>");
  curr_radio.insertAfter(legend);
  curr_radio.after("<br/>");
}
empty_radio_row.remove();
legend.after("<br />");
```

```
    });

});
});

});
```

3 Preview

The third subsection of Designing job is Previewing the Works you have done in second subsection. In our work, we asked workers to identify and reply two questions for each spot with the following guideline: For each tweet you have to do two jobs: Select the appropriate option by understanding the tweet meaning and eventually the meaning of the words in green color Rate how much your choice is relevant.

Steps: Read the tweet (more carefully the words in green color) Click on "view in Wikipedia" of each option Select the option which has the proper wiki page according to the meaning you got from tweet If none of the options had the appropriate wiki page, please select "None of them" and give your motivation in the box appearing below. (In this case step 5. is not required) Finally, select how much your choice is related according to the meaning you got from tweet .

So the questions will appear for contributors as the following image:

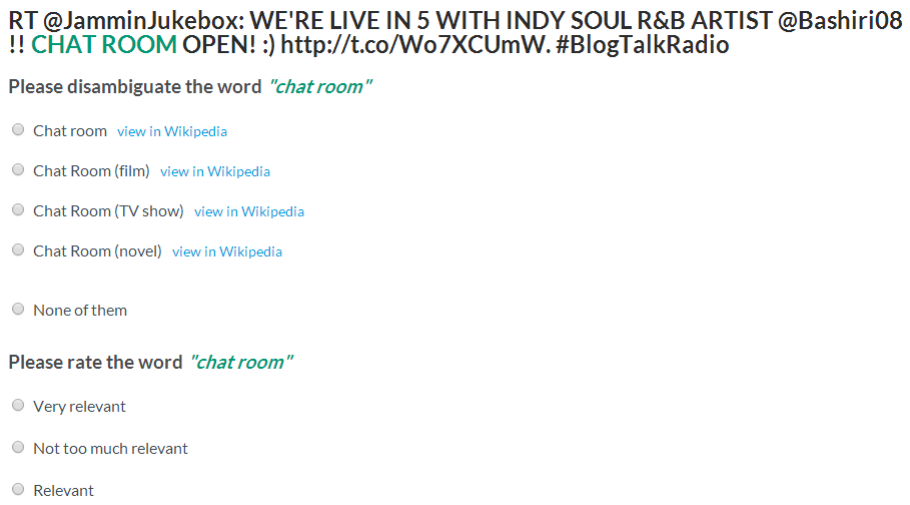


Figure 4: One of Our Questions

We can use Insert Data to insert Liquid variables in the form. The variables shown in the dropdown menu correspond to the column headers in your dataset. CSS Field Click the Show Custom CSS/JS link to access the CSS field. This field will allow you to add custom styling to your assignments.

Figure 5: CSS

. This field allows you to add any custom JavaScript you need to your

form.

• It is highly probable that some users use the Graphical Editor since it is easy to work with it, but eventually when they need to add some more complicated features they want to change for CML editor, this action is not supported from Crowdslower to keep all the data, so if a user in the middle of work change the editor from Graphical Editor to CML editor or vice verse, she will lose data.

Consider User Experience

• Minimize scrolling & clicking: Forms with lots of moving parts like scroll bars and buttons can become confusing. Carefully consider form layout to reduce worker fatigue and improve efficiency.

• Keep it local: Whenever possible, avoid asking contributors to navigate away to external sites to complete a task.

• Provide shortcuts and hyperlinks: If your job does require navigating to other webpages or searching the internet, make sure you support contributors with shortcuts, such as hyperlinks to a predefined google search in the following format: http://www.google.com/search?q=some+query. When using our custom Liquid validator you can use the following format to encode an unknown piece of data: http://www.google.com/search?q={{ your_data | urlencode }}. You can also create a link with an HTML tag in the following way: <a href="http://www.google.com/search?q=your_query" target="_blank" />.

• View your job through the contributor's eyes: Use CrowdFlower's Preview tab to see the first few units of your job. Check the logic on your form through the Gold Creation UI, or use the following URL structure to preview your entire job just as contributors will experience it:

http://crowdflower.com/jobs/[YourJobID]/preview.

Part II
Manage Quality

4 Test Questions

Second step is Managing Quality that this step also has three subsections as well, the first one is creating Test Questions. Test Questions also known as Gold set, are units with known answers that are regularly inserted in the job.Test questions has a significant rule in improvement of results. Test questions can train contributors by showing them that why they got the

test question wrong in case that they fail in answering correct; consequently contributor can understand her error and fix it in the following questions. The other benefit of test questions is removing underperformers, when some contributors fail too many test questions, system remove them and all of their answers from the job. Definitely we can say that test questions are the most important quality control mechanism in the CrowdFlower Platform. Actually Test Questions are units that requester has already known the answers and they are inserted in the job quite randomly, therefore there is also no way to cheat by a contributor. These are useful also to test and track contributor performance.

By using Test Questions, requester is completely sure that she will get results only from trusted contributors who are proved their competency for doing her job. Test Questions are used two times, once in Quiz Mode and before a contributor enter to a job, and the other time during the job. In this way a requester can understand the performance of each contributor that works on a job and be sure that contributors with poor performance will automatically be removed. Moreover there is an option in Crowdflower that requester can set a threshold of accuracy in a job, then CrowdFlower Platform will only allow to some contributors to work on a job that they already have performed above that threshold. A very important issue in preparing Test Questions is that requester should explain about the reason of selecting an answer to a Test Question. As these answers will be shown to contributors when they get a Test Questions incorrect, so it's vital for them to know their errors then they may learn and understand how the task performance is. Another critical point that a requester should pay attention to it, is that the Test Questions reasons should be clear and instructive. Creating Test Questions Creating Test Questions is the process of going through your source data and marking acceptable answers for contributors to be tested on when you run your job. The easiest method for creating Test Questions is Creating Test Questions in the Platform and for this matter there is an easy-to-use interface in Crowdflower. Any way creating Test Questions procedure is the following: After being sure that the CML form is completed and the data is uploaded then by clicking on the Test Questions tab in the Platform, then clicking on the blue button labeled "Create More" on the right side of the page. As a result going on through each of the units, and selecting those of which you want them to use as Test Questions.

At the end requester should answer those that she has already selected them as Test Questions correctly then provide a reason why that answer is the correct one. For providing a Test Question reason, requester should mention how the correct answer was found and tips on how to identify the

correct answer. In this way crowd can be taught how to provide the best responses to the requester. So the biggest advantage of doing is effort is for requester by getting best results. If there are some units that would not make a good Test Question because for example they have difficult subject, requester can skip them by clicking the Skip button.

Manual Test Questions :

Requester can create a lot of Test Questions via a spreadsheet and upload all of them at the same time. Requester can do it either by uploading a Test Question report or Include Test Questions in the Source Data. It is possible to upload a separate report that only contains Test Questions which are directly in the Test Question interface. Select the Reports drop down in the Test Questions menu and select Upload Test Question Report. To accomplish this: First we should have an appropriately formatted report; we can do it by creation a fake Test Question and then download the Test Question Report. As a result each CML element will contain two more columns that will store the value and reason of Test Question

It is also possible to upload Source Data that contains the Test Questions by including the Test Question columns in the Source Data. In Graphical Editor it is done by activating the "Mark as gold" checkbox on each form element and in CML editor, by adding the attribute gold="true" to the CML elements. Providing that the Source Data must contain the Test Question value and reason columns for this method to work. The source data should have a column with name _golden and for the Test Questions this should be "TRUE". In the data page, after uploading the source data containing the Test Question values, by selecting the Manage Data button and then clicking on "Convert Uploaded Test Questions", the units with Test Questions value and reasons will turn to Test Questions.

Besides Test Questions can have multiple correct answers. It occurs when the response of contributor is more subjective than objective. When specifying Test Questions manually, multiple values in the Source Data will be determined by the newline character \n.

5 Contributors

In Contributors subsection of Manage Quality step, you can select your contributors as you wish, by defining or limiting their geography situation, their skills including their performance level which has been already defined by crowdflower for each worker, their language capability, their behavior setting which allows you to set the maximum amount of judgments a contributor

can submit on a job. In addition you may also define Max Judgments Per Contributor it means number of judgments that a certain contributor can do in your job and it's according to each contributor's ID. This feature is used when a requester does'n't want to do risk with getting a high number of judgments from a contributor which is weak also using this setting make the requester sure that no single contributor can see a Test Question more than once. Using Max Judgments per Contributor is more important when a job has less than a few hundred Test Questions. It is useful to avoid cheating or answer sharing on Test Questions. • Max Judgments Per Contributor is calculating by multiple of the amount of Units per Task and the amount of Test Questions in a job, divided by the number of Test Questions per Task. Moreover since an individual contributor submits only one judgment per unit, the number of Units per Task is the same as the number of judgments per task.

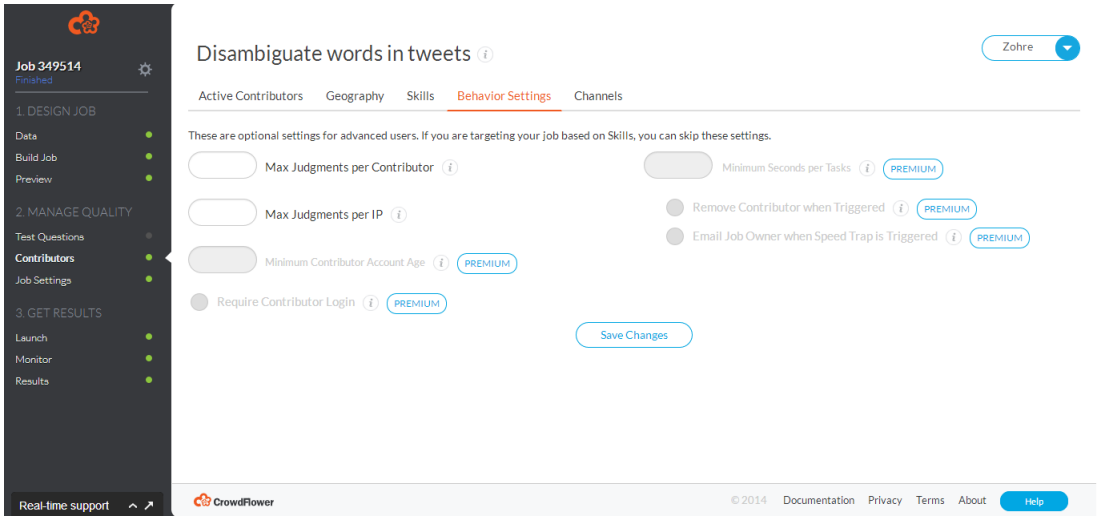


Figure 6: Behavior Settings

There is other feature named Max Judgments per IP, it is so similar to Max Judgments Per Contributor with the difference that will affect to stop contributors from cheating, because if there was no Max Judgments per IP, one contributor can have several username and by attempting several times, they will be able to recognize the Test Questions and easily cheat. Also Max Judgments Per IP is calculated with the same method of calculating Max Judgments Per Contributor. The other features are specified for Premium users of Crowdflower, ones that already were active and did certain amount

of jobs.

Effectively Crowdflower system uses to calculate Max Judgments Per IP as max per contributor multiples by two. This is used for some contributors which are working from the same Computer in stance a husband and wife or siblings team. Also we can define Min Seconds Per Task which is the time that a contributor must spend on a task. This is used for being sure that contributor put enough time to answer the question. Likely the speed trap feature is the minimum time that a contributor should spend to complete one page of work. It's for being sure that a contributor is paying enough attention in the job and this is specific for premium requesters.

6 Job Settings

6.1 Tasks

In this subsection requester can decide and define few thing for instance she can decide about the price to pay to contributors for doing the job, the number of units that a contributor should do for completing a task and the time that a contributor has on a task. Requester should keep in mind that the more work for completing a task requires more money, but in deed we relied on Crowdflower automatic decision about the price, and it was almost 7 dollar cents for each task included three units. For deciding about number of units per task, Test Questions should be considered. As we know Test Questions will be inserted randomly. So when requester wants to have two units for each task, for number of Units per task, requester should set 3 instead of 2. As a result in each task contributors will answer two questions from source units and one from Gold units or in other word Test Questions. For sure contributors will be notified if they miss the Gold unit hidden within the page. Moreover when a requester has Professional account, from the Test Questions Settings page, she can change the number of Test Questions Per task. In the Task Expiration Time, requester can define an expiration time and it is highly recommended to set long time than expected. In this case requester considers about emergency matters which is estimated to be occurred during a work.

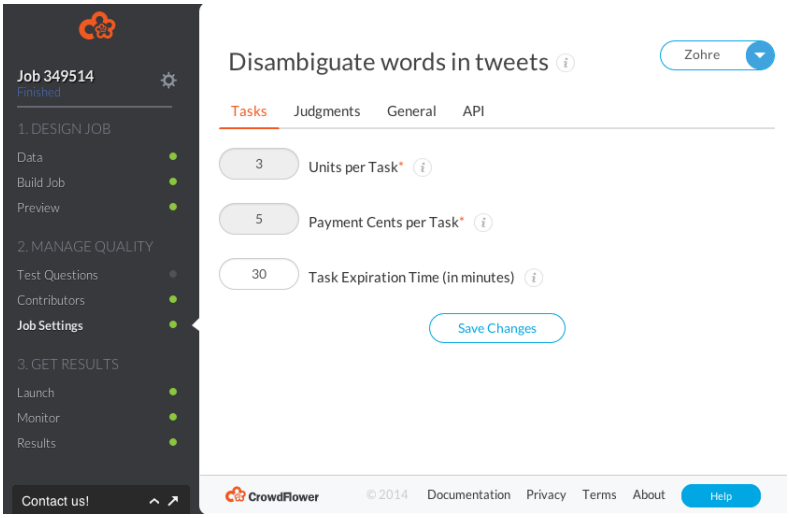


Figure 7: Job settings

6.2 Judgments

This page is important to decide about the job and the way it will be judged. Obviously having multiple judgments will make the requester sure about finding the proper answer. Although the response of trusted contributors are so reliable, considering a contributor as a human who is able to do mistake in any time, Crowdflower prefers to have multiple contributors to judge about the same job. Before a unit is finalized, there should be a minimum number of trusted judgments for each unit and this is settable in Judgments per unit part of the page. According to the value that we put in aggregation attribute in the CML form elements, these judgments will be aggregated on the backend. In some kind of jobs, it is important for requester to have answer in order that she has uploaded them, in this case she should activate this article in job setting page, which is Units should be completed in the order they were uploaded. The default for this matter is in random order; means that units appear in a task in random order.

6.2.1 Variable judgments mode

There are some cases that even though there is done the number of judgments that requester has already set the number of them, in case that confidence on one or more of task field is less than the threshold or in other words, minimum confidence. By enabling “Variable Judgments Mode”, when the

cases which was talked above happens, there will be additional judgments. In this condition, by enabling this option, there will be more setting to do:

Max judgments per unit; Again there will be maximum number of judgments that a unit will receive and it will stop even if the confidence on the unit will not be more the minimum confidence. It is very important to use this to prevent to have thousands of judgments and wasting a lot of money in case that a unit has problem and doesn't approve by contributors. "Expected judgments per unit" which is the number of judgments that requester expects that units will receive on average. This is useful for platform to know the number of judgments that is necessary to complete the job. It is calculated by expected judgments per unit multiplied by number of units.

Minimum confidence Units that fall below this confidence (agreement between contributors) on the confidence fields you select (see Confidence fields below) will receive more judgments. The unit will continue collecting judgments until either the minimum confidence or the max judgments per unit is reached. Stop below confidence Units that fall below this confidence (agreement between contributors) on the confidence fields you select (see below), will not receive additional judgments. These units are typically the most difficult or problematic units in your dataset. Confidence fields These are the fields that Variable judgments will be monitoring. Confidence on these fields will determine if additional judgments are necessary based on your Variable judgment settings.

• Variable judgments occur on the unit level when the confidence on one or more these fields falls below the minimum confidence.

• These fields are listed based on the "name" or "label" attributes in the CML elements.

Only the cf_sentiment CML element is selected to collect variable judgments in this job. The relevance CML element is not selected and will collect the judgments per unit set as it will not collect variable judgments.

6.3 General

With this article a requester can decide about the way for getting informed of progression of the job. In "Notify" by writing one or some email addresses, notification from platform will be sent, specially one message will be sent when the job gets complete. In "tags" requester can type some keyboards to help people to find the job for instance about the kind of job or the name of team, etc. which are metadata associated with jobs, jobs will be available in the overview section of the platform. In addition by enabling the "Make your data public" subsection, the data which was processed in the job can be

usable for public in CrowdFlower Open Data Library Knowing that enabling this subsection will force the requester to agree with Customer terms and Conditions.

6.4 API

This article describes the settings for how a job will communicate with the API and react to API commands. First of all it is better to say that a webhook is an address or in it the other word URL of the server or machine that a requester wants to control her job by getting or sending messages. So by writing a web address requester can automatically get important messages about the job. This is more useful for requester who wants to get instant results about the job. Also this procedure will be expired by finishing the job. By webhook, requester can get updates when unit complete, job complete or job data processed. Each of these signals will be accompanied with a JSON (JavaScript Object Notation) payload. Alternatively JSON objects can be loaded with JS (on page load), if the value is written to a form element (as a string) using Liquid.

Part III
Results:

The most interesting part of this work, is getting result after very short time which took only 71 minutes to finish all. For sure this is the most important advantage of Crowdflower since it distributes the work on 50 labor channel besides it has over 5 million contributors. After lunching the job and after finishing the work which for us was an hour and 11 minutes. As we already wrote, in this work we have 208 units, each unit, in each task we put three units, and we ask three times of judgment for each task, so at the end we had 208*3= 624, of course for each task we get judgments from three different contributors. In the “result ” section of our project, there are five reports in the format of .CSV

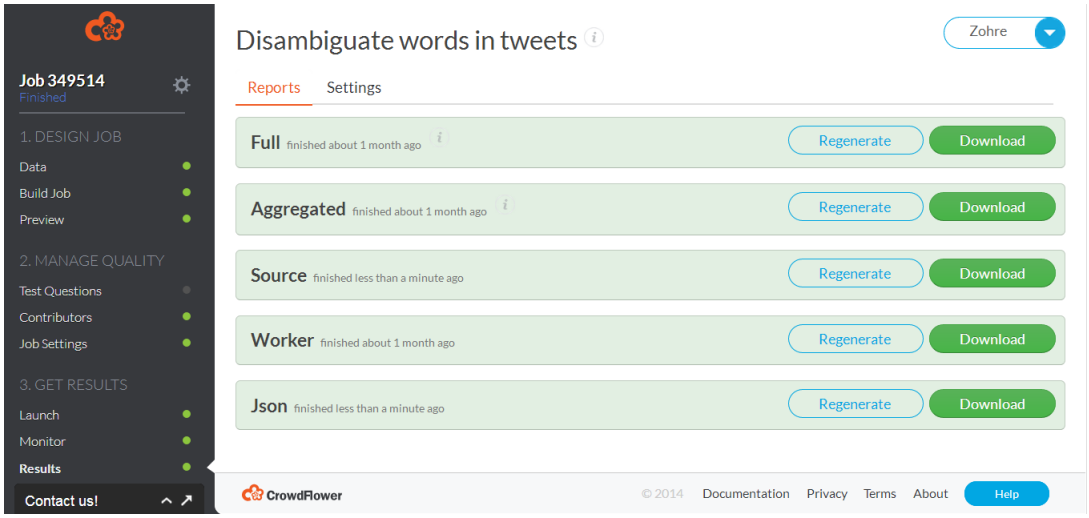
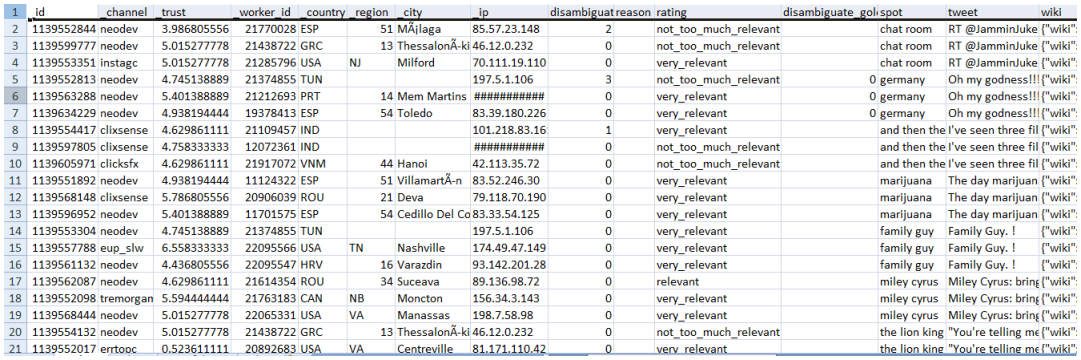


Figure 8: Results

to generate them and download. Two of them is so important and they are: "all" and "aggregated" named reports. First report is named "full" by Crowdflower which returns all unique responses submitted by all trusted contributors for the given field. The result will be a newline '\n' delimited list in the Aggregated report. "Full" named report contains the information of all 624 judgments with enough information about each of them. For example each judgment which here in this file is represented as a row has different features for example each row has a unique Id, the tweet, the spot, the possible and suggested options to be voted, her answers to it, both for the disambiguation and relevance level, besides unit_id is the same for a certain unit, so there is one unit_id for every three units because number of judgments are three. Moreover we can know that in which channel a judgment was done. This one is important for future analysis, since if we find out that judgments from certain channel is poor, so in the later effort we can limit that channel. Also we can see that any judgment was done from which country, region, city and even IP.



	id	_channel	_trust	_worker_id	_country	_region	_city	_ip	disambiguat	reason	rating	disambiguate_gol	spot	tweet	wiki
2	1139552844	neodev	3.986805556	21770028	ESP		51 MÃ¡laga	85.57.23.148	2		not_too_much_relevant		chat room	RT @JamminJuke	{"wiki":
3	1139599777	neodev	5.015277778	21438722	GRC		13 ThessalonÃ-ki	46.12.0.232	0		not_too_much_relevant		chat room	RT @JamminJuke	{"wiki":
4	1139553351	instagc	5.015277778	21285796	USA	NJ	Milford	70.111.19.110	0		very_relevant		chat room	RT @JamminJuke	{"wiki":
5	1139552813	neodev	4.745138889	21374855	TUN			197.5.1.106	3		not_too_much_relevant	0	germany	Oh my godness!!!	{"wiki":
6	1139563288	neodev	5.401388889	21212693	PRT		14 Mem Martins	###########	0		very_relevant	0	germany	Oh my godness!!!	{"wiki":
7	1139634229	neodev	4.938194444	19378413	ESP		54 Toledo	83.39.180.226	0		very_relevant	0	germany	Oh my godness!!!	{"wiki":
8	1139554417	clixsense	4.629861111	21109457	IND			101.218.83.16	1		very_relevant		and then the	I've seen three fil	{"wiki":
9	1139597805	clixsense	4.758333333	12072361	IND			###########	0		not_too_much_relevant		and then the	I've seen three fil	{"wiki":
10	1139605971	clicksfx	4.629861111	21917072	VNM		44 Hanoi	42.113.35.72	0		not_too_much_relevant		and then the	I've seen three fil	{"wiki":
11	1139551892	neodev	4.938194444	11124322	ESP		51 VillamartÃ-n	83.52.246.30	0		very_relevant		marijuana	The day marijuan	{"wiki":
12	1139568148	clixsense	5.786805556	20906039	ROU		21 Deva	79.118.70.190	0		very_relevant		marijuana	The day marijuan	{"wiki":
13	1139596952	neodev	5.401388889	11701575	ESP		54 Cedillo Del Co	83.33.54.125	0		very_relevant		marijuana	The day marijuan	{"wiki":
14	1139553304	neodev	4.745138889	21374855	TUN			197.5.1.106	0		very_relevant		family guy	Family Guy. !	{"wiki":
15	1139557788	eup_slw	6.558333333	22095566	USA	TN	Nashville	174.49.47.149	0		very_relevant		family guy	Family Guy. !	{"wiki":
16	1139561132	neodev	4.436805556	22095547	HRV		16 Varazdin	93.142.201.28	0		very_relevant		family guy	Family Guy. !	{"wiki":
17	1139562087	neodev	4.629861111	21614354	ROU		34 Suceava	89.136.98.72	0		relevant		miley cyrus	Miley Cyrus: bring	{"wiki":
18	1139552098	tremorgan	5.594444444	21763183	CAN	NB	Moncton	156.34.3.143	0		very_relevant		miley cyrus	Miley Cyrus: bring	{"wiki":
19	1139568444	neodev	5.015277778	22065331	USA	VA	Manassas	198.7.58.98	0		very_relevant		miley cyrus	Miley Cyrus: bring	{"wiki":
20	1139554132	neodev	5.015277778	21438722	GRC		13 ThessalonÃ-ki	46.12.0.232	0		not_too_much_relevant		the lion king	"You're telling me	{"wiki":
21	1139552017	errtopc	0.523611111	20892683	USA	VA	Centreville	81.171.110.42	0		very_relevant		the lion king	"You're telling me	{"wiki":

Figure 9: Full report

The other file named “aggregated” which the most useful information is in it, which returns a single "top" result - AKA the contributor response with the highest confidence (agreement weighted by contributor trust) for the given field. All other responses will be ignored. So it works such that in the base of confidence of each judgment, it selects one judgment from three of them, indeed it is selected the one that its contributor has high confidence from Crowdflower.

	unit_id	rating:confiden	disambiguate:confider	trusted_	judgm	reason	rating	spot	tweet	wiki
2	366268269	1	0.5926		3	asdfasdfasdf	very_relevant	kolob	RT @billmaher: Mi	{"wiki": [{"prior": 0.9523809523809523, "id": 670655, "title": "Kolob"}
3	366268270	0.5062	0.3737		3	He is just saying only	not_too_muc	only you	RT @bereolaesqu	{"wiki": [{"prior": 0.22043010752688172, "id": 1296348, "title": "Only
4	366268294	1	0.6474		3	dfvgfdvgfdgThe tweet is	very_relevant	only you	All I need is only yc	{"wiki": [{"prior": 0.22043010752688172, "id": 1296348, "title": "Only
5	366268329	1	0.6649		3	noneNo, porque esta ha	relevant	old enough to	RT @iTweetYouLo	{"wiki": [{"prior": 0.5882352941176471, "id": 16295805, "title": "Old_
6	366268357	1	0.7297		3	play the videoits none o	not_too_muc	press play	#certifiedbanger fr	{"wiki": [{"prior": 0.8648648648648649, "id": 5959386, "title": "Press
7	366268369	1	0.6757		3	big bang theoryit is tv sl	not_too_muc	big bang	I'm watching The E	{"wiki": [{"prior": 0.7131072410632447, "id": 4116, "title": "Big_Bang
8	366268394	1	0.7297		3	Thatjust a name of the s	very_relevant	shrimp	Organic Baby, Kid	{"wiki": [{"prior": 0.8671023965141612, "id": 36762240, "title": "Shri
9	366268400	1	0.7123		3	it means now lall those	very_relevant	na i	Ol na I hate two fa	{"wiki": [{"prior": 0.46153846153846156, "id": 30528129, "title": "Na
10	366268403	1	0.6087		3	He is telling a girl sorry I	not_too_muc	take that	I hope Brittani doe	{"wiki": [{"prior": 0.9929245283018868, "id": 371370, "title": "Take_T
11	366268405	1	0.7027		3	The people in said twee	not_too_muc	up all night	RT @FunnyEvil: I'r	{"wiki": [{"prior": 0.2764423076923077, "id": 31756817, "title": "Up_/
12	366268407	1	0.6721		3	asasdfasdfasdfThis twe	relevant	love sosa	bitches love sosa	{"wiki": [{"prior": 1.0, "id": 37666146, "title": "Love_Sosa"}]}
13	366268413		1		3	It's none of the above because it was meant as a name for his speech. its part of obamas speech		the best is ye	RT @AnnCoulter:	{"wiki": [{"prior": 0.6464646464646465, "id": 9865624, "title": "The_E
14	366268414	1	0.6842		3	He meant drawing as in	very_relevant	stupid shit	RT @OMGJq: Wh	{"wiki": [{"prior": 1.0, "id": 17167704, "title": "Stupid_Shit"}]}
15	366268418	1	0.6481		3	boii is a different way of	very_relevant	boii	RT @blunt_blowin	{"wiki": [{"prior": 0.9798657718120806, "id": 4327, "title": "Boii"}, {"p
16	366268426	1	0.6213		3	relevantsdfasd	very_relevant	barack obam	Redskins Rule: Ba	{"wiki": [{"prior": 0.9916494845360825, "id": 534366, "title": "Barack
17	366268441	1	0.6875		3	Doesn't mean anything	very_relevant	home alone	RT @cookielanga	{"wiki": [{"prior": 0.8356643356643356, "id": 216072, "title": "Home_
18	366268461	1	0.7022		3	Because it is clear the c	very_relevant	marietta	I Need To Go To M	{"wiki": [{"prior": 0.525887573964497, "id": 109987, "title": "Marietta

Figure 10: Aggregated report

Confidence value is an integer in range of 0 to 1. This report has unit _id which for each unit is unique, spot, tweet, the possible and suggested options to be voted, and a numerical confidence value which is in range of 0 to 1. More importantly this report is included the final vote which is selected from three judgments. In our case, it gives back the Wikipedia page which is selected by crowd and also its relevance. By analyzing these two reports,

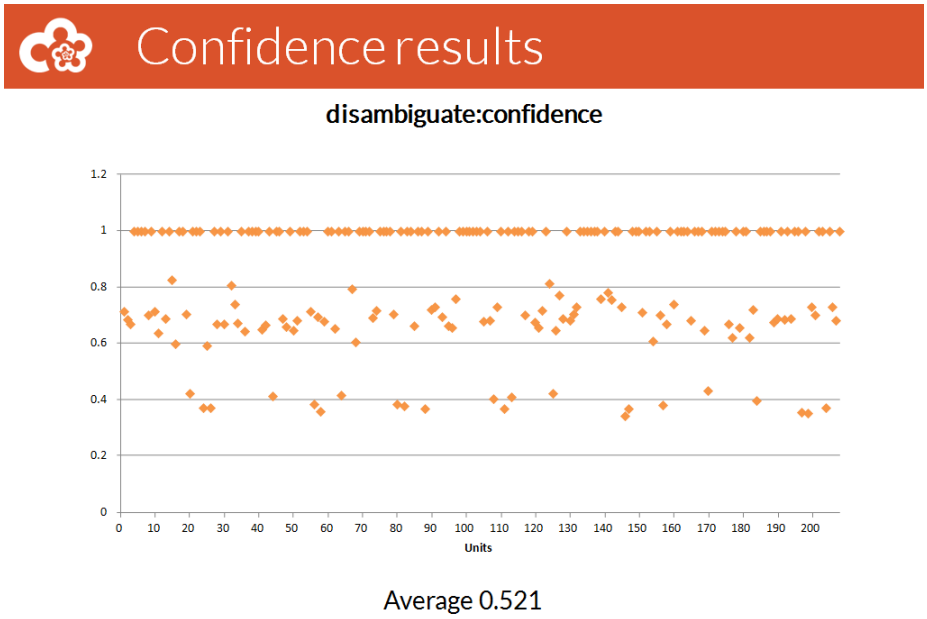


Figure 11: Disambiguation Confidence

we can reach to some important results.

From this chart, we can see that the average of Disambiguation confidence is 0.521. It's totally depend on the requester that how to decide about discarding the judgments by putting a threshold in confidence. In our case, after analyzing the information, we found out that units with less than 0.5 means that all the three judgment's results are different, for example for a certain question, worker 1 voted for A, worker 2 for B and worker 3 for C, and at the aggregated report the one that it's worker has high confidence from Crowdflower, is selected. Therefore in our work we decided to discard the votes with less than 0.5 in disambiguation confidence. Fortunately their number is not so high, from 208 units we have just 22 of them with less than 0.5 disambiguation confidence which is just almost 10% of all the units. Instead there are 109 units with Disambiguation confidence of 1, it means that all three judgments for a unit was the same and it covers 52% of all the units. In fact this kind of report is the most reliable for us, since all the contributors are agreed and compromise in one answer. Number of the remain part which their disambiguation confidence id between o.5 and 1, is 77, which is 37% of all and we rely on these judgments too.

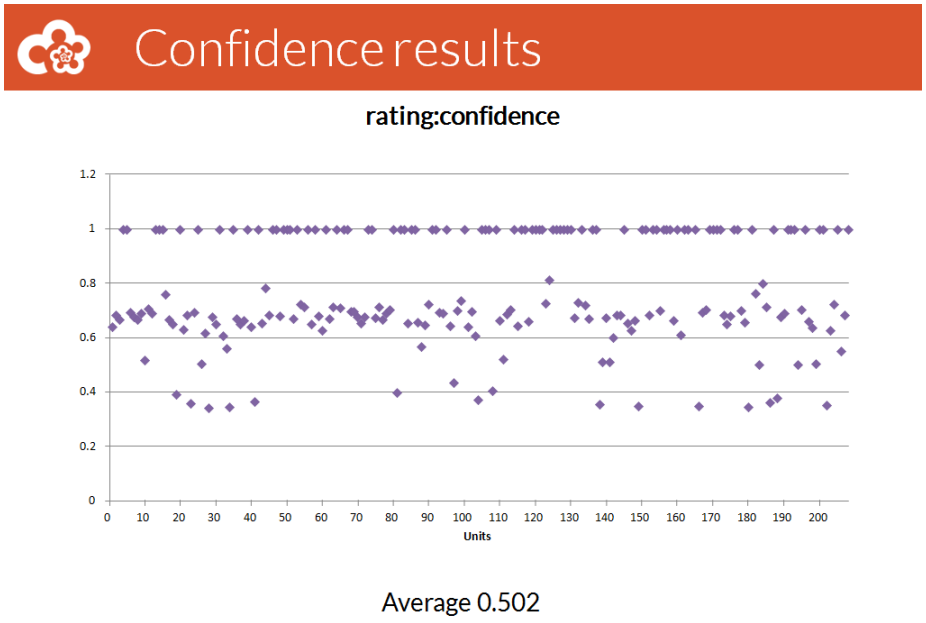


Figure 12: Rating Confidence

In comparison the average of rating confidence is 0.502.

Analyzing answers of second part of question which is "rate" part, we find out that like the disambiguation confidence, also here, the number of units with rate confidence less than 0.5 is not so high, even if we withdraw them, it will not be such a big problem. On the contrary of disambiguation confidence, here the number of units with rate confidence between 0.5 and 1 is more than number of them with rate confidence of 1.

Since there is no grantee about the rules for writing tweets, in the first appearance it seemed that there would be a lot of units with answers of Not relevant, but in fact at the end we reached to this statistic :

It is so important that at the beginning how the requester represents the Instruction and also the reason of selecting proper answer in Test Questions. If this step goes well, there will be limited number of units which contributors get misunderstanding with them. For instance in our job, according to contributor's idea, our instruction was 78% clear for them. There are other reasons to decrease the confidence too. In our reports the unit with lowest disambiguation confidence which is 0.3432 and means that it is completely misunderstood by contributors. Here is the unit(tweet) that is the most

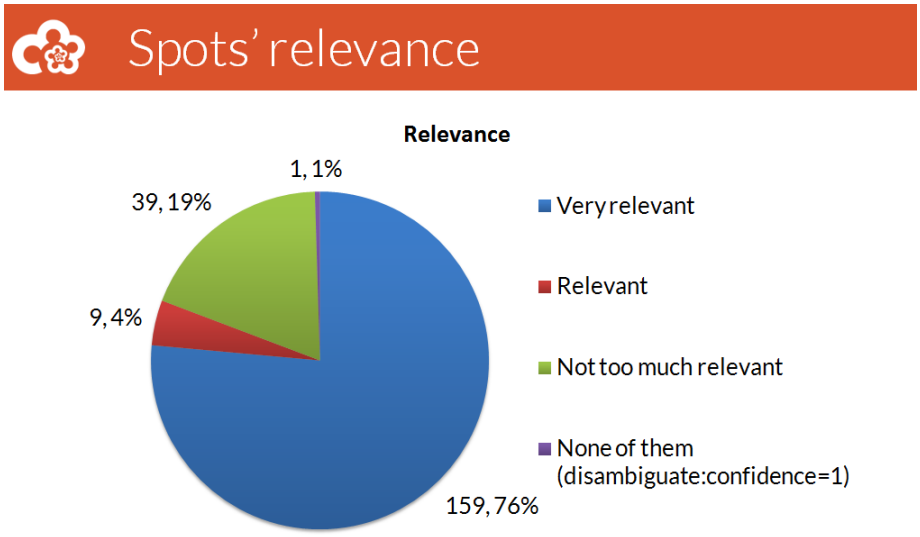


Figure 13: Spots' relevance

misunderstood by contributors:

> The worst nicknames in sports - The "Muscle Hamster", "Molester", and "Flying Tomato"? http://t.co/S9R79kL5

For this unit the disambiguate confidence is 0.3432, the rating confidence is 0.6568 and there are 5 Wikipedia pages to be voted.

Channel	Trust	Worker ID	Country	City	Rating	Wiki
neodev	4,94	19378413	ESP	Toledo	not_too_much_relevant	{"prior": 0.9490, "title": "Tomato"}
neodev	4,71	21432512	TUN		very_relevant	{"prior": 0.0093, "title": "Tomato_(company)"}
neodev	4,75	21374855	TUN		very_relevant	{"prior": 0.0079, "title": "Tomato_(musician)"}

Table 1: Lowest disambiguate confidence

On the other hand by analyzing the report we find out that the worst(least) rate confidence is 0.3447 with disambiguate confidence of 0.6698 and 7 Wikipedia options.

The tweet is:

> RT @6CancerZone9: No secrets are allowed to be kept from a #Cancer. Thats our job.

The"Lowest rating confidence" table is more information about this tweet.

Channel	Trust	Worker ID	Country	City	Rating	Wiki
neodev	4,94	19378413	ESP	Toledo	relevant	{"prior": 0.5125, "title": "No_Secrets _(girl _group)"}
errtopc	0,52	20892683	USA	Centreville	very_relevant	{"prior": 0.5125, "title": "No_Secrets _(girl _group)"}
neodev	5,01	22065331	USA	Manassas	not_too_much _relevant	{"prior": 0.0125, "title": "List_of _Trinity _episodes"}

Table 2: Lowest rating confidence

There is no limitation about analyzing the reports and finding out interesting information, which is absolutely useful for our next works, consequently by changing and modifying the parts that we already got their weakness from previous times, we will reach to more useful achievements.

In our case, it is also important to know how many spots are voted “Not too much relevant ”:

The following are tweets with judgment of “Not too much relevant”

> RT @HaydenIsaiah: Don’t act like you like President Obama now since he’s President and you voted Mitt Romney! Mitt Romney was gone have ...

For this tweet the disambiguate confidence is 1, rating confidence is 1, and it is with two wikis:{"prior": 0.9592592592592593, "id": 426208, "title": "Mitt_Romney“}

The other tweet is:

> Lol got me RT @_theveroniKa: @J_Hardaway okay omarion lol

With these information: disambiguate:confidence = 1, rating:confidence = 1,With only one wiki {"prior": 1.0, "id": 186260, "title": "Omarion”}

One more tweet with this condition is:

> RT @JustineLavaworm: For those saying "if Obama wins I'm going to Australia" our PM is a single atheist woman & we have universal he .

With disambiguate confidence = 0.6467, ratingcconfidence = 1, two wikis {"prior": 0.9652032520325203, "id": 15247542, "title": "Atheism"} (2 votes) {"prior": 0.03219512195121951, "id": 526797, "title": "Atheist_(band)"} (1 vote)

There is a decent point, in our data there were some spots with only one Wikipedia pages, in first appearance, it was so odd to ask crowd to judge a single option question, but the point is that since we have the option of "NONE", which every contributor can select it when she couldn't find the appropriate one among the suggested options. Knowing this make the judgments for single Wikipedia pages sensible.

Part IV
Conclusion

In this thesis, we took a set of tweets, where some subsequence of words or in the other word spots were annotated with possible meaning/entities and these spots were linked with Wikipedia pages. Then we created a crowd-sourced system in Crowdflower in order to study user behaviors, evaluate the outcome and discuss the results. We investigated that Crowflower system is the best for our work. We designed our job in Crowdflower Markup Language (CML). As Crowdflower uses CML and Liquid to generate the form needed for each unit of work. The same CML code is used for all units in the dataset. Also Crowdflower allows us to use JS (w/ jQuery) and CSS to further customize the CML. We uploaded 208 tasks of data; each task was included three units. Each unit was one question asking user the correct Wikipedia page linked by the defined word/spot and rate for it considering how much the same spot is relevant for understanding the essential meaning of the tweet. We repeated this job three times. After 72 minutes, Crowdflower submitted us answers of all questions with the full report of all works done by crowd. Since we got three answers for each question, Crowdflower selected the best answer and show to us. At the end we know that they are just reports and finally we ourselves should decide which answers should be considere as spam. Thus we decided to withdraw outputs with Confidence

less than 0.5 since these kind of outpute were the ones with three different answers, however Crowdflower approved one whose user has high trust_id. In the other hand majority of our outputs, both in disambiguation and rating phase, was with confidence 1 which means all three answers were the same for them.

References

[1] Wikipedia, http://en.wikipedia.org/wiki/Crowdsourcing

[2] G Kazai, In Search of Quality in Crowdsourcing for Search Engine Evaluation, 33rd European Conference on IR Research, ECIR 2011, Dublin, Ireland, April 18-21, 2011. Proceedings

[3] J. Le, A. Edmonds, V. Hester, and L. Biewald. Ensuring quality in crowdsourced search relevance evaluation. In Proceedings of the ACM SIGIR 2010 Workshop on Crowdsourcing for Search Evaluation (CSE 2010), pages 17–20, Geneva, Switzerland, July 2010.

[4] J. S. Downs, M. B. Holbrook, S. Sheng, and L. F. Cranor. Are your participants gaming the system?: screening mechanical turk workers. In CHI '10: Proceedings of the 28th international conference on Human factors in computing systems, pages 2399–2402. ACM, 2010

[5] A. Kittur, E. H. Chi, and B. Suh. Crowdsourcing user studies with Mechanical Turk. In CHI '08: Proceeding of the 26th SIGCHI, pages 453–456, 2008.

[6] Omar Alonso , Daniel E. Rose , Benjamin Stewart, Crowdsourcing for relevance evaluation, ACM SIGIR Forum, v.42 n.2, December 2008

[7] O. Alonso and S. Mizzaro. Can we get rid of TREC assessors? Using Mechanical Turk for relevance assessment. In Proceedings of the SIGIR 2009 Workshop on the Future of IR Evaluation, pages 15–16, 2009.

[8] G. Kazai and N. Milic-Frayling. On the evaluation of the quality of relevance assessments collected through crowdsourcing. In

Proceedings of the SIGIR 2009 Workshop on the Future of IR Evaluation, pages 21–22, 2009.

[9] Tim Finin , Will Murnane , Anand Karandikar , Nicholas Keller , Justin Martineau , Mark Dredze, Annotating named entities in Twitter data with crowdsourcing, Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, p.80-88, June 06-06, 2010, Los Angeles, California

[10] Omar Alonso , Ricardo Baeza-Yates, Design and implementation of relevance assessments using crowdsourcing, Proceedings of the 33rd European conference on Advances in information retrieval, April 18-21, 2011, Dublin, Ireland

[11] Kazai, G., Kamps, J., Koolen, M., Milic-Frayling, N.: Crowdsourcing for Book Search Evaluation: Impact of HIT Design on Comparative System Ranking. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information, pp. 205–214 (2011)

[12] Crowdflower, http://crowdflower.com/