Human Computation and Crowdsourcing: Works in Progress and Demonstration Abstracts
AAAI Technical Report CR-13-01

GameLab: A Tool Suit to Support Designers of Systems with Homo Ludens in the Loop

Markus Krause
Leibniz University Hannover, Germany
markus@hci.uni-hannover.de

Abstract

Digital games are an interesting method to motivate contributors to part take in a human computation process. However this approach poses its own challenges. Especially quality management or immediate and diverse feedback to players are recurrent challenges. This paper introduces a tool suit to support designers with these challenges.

Introduction

Most human computation systems share a common structure for obtaining the desired results. A *requester* has a *task* that is currently too difficult to automate. They break this task into small manageable pieces called *requests*. These requests are distributed to contributors who respond to them and the system aggregates one or more *responses* into *answers*. In this paper we present a tool set that helps to handle the common challenges of such an endeavor.

GameLab Tool suite

The *GameLab*[1] tool suit is designed to support game designers with the following tasks:

- distribute requests to contributors
- aggregate responses given in natural language[2]
- rate responses
- provide diverse and useful feedback to players

GameLab uses a set of methods to estimate and ensure response quality. A simple feature used is to restrict the number of responses per contributor per request. This way no single contributor can pollute the database. It uses *wordnet* (Miller, 1995) to detect swear, harassment, and slang words. G*ameLab* detects overly frequent use of terms (called *fixation* in the algorithm). It compares the term frequency of terms from the contributor with the term frequency in the whole task. Most important *GameLab* uses a semantic similarity measurement for comparing two responses via *wordnet* instead of string based comparisons[3].

GameLab identifies misspelled words and detects random strings with *Language Tool*[4]and *wordnet*. It also tracks the quality of responses given by a contributor over time. If no assumption on the quality can be made with the previously described methods. This time series is used to estimate the response quality. Based on these methods *GameLab* builds a feature vector for every response. It calculates the vector in real-time (without a noticeable delay) which is an important factor in interactive scenarios such as games. From this vector the system also calculates a trust value that represents the quality of a response. The algorithm to calculate this value is shown in Figure 2. The feature vector and the trust value are stored in a feedback object. This object is then sent to the system submitting the response.

Experiment

We conducted an experiment to shed light on the question: whether the feedback of *GameLab* can influence response quality? To answer this question we published two prototypical games. *GuessIt* and *Empathy* both games share similar game mechanics and the same data set. The dataset used for the experiments consist of ~3600 images. The first game is *Empathy*. Figure 3 shows a screenshot of the game. The player enters a label in the text field below the image. *Empathy* uses *GameLab* only to distribute and aggregate responses but not use the response evaluation methods. The system calculates the score for a response based on labels already in the database. Labels for an image in the database are ranked based on their frequency. If an entered label matches the most frequent label the score is three. If the label matches any other label the score is two.

[1] https://code.google.com/p/gamelab/
[2] GameLab currently supports English only.

[3] GameLab uses the ws4j implementation of the semantic distance algorithm from Wu and Palmer (1994). (https://code.google.com/p/ws4j/)
[4] GameLab uses version 2.1 (http://www.languagetool.org/)

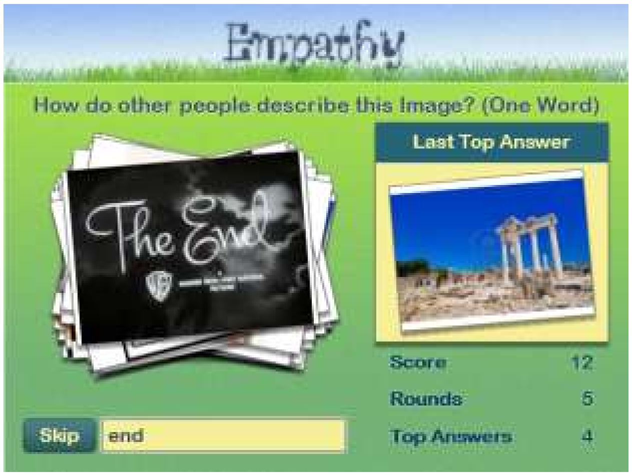

Fig. 3. *Screenshot of the* Empathy *game.*

If no term exists the score is determined by the scores of the last 10 responses divided by 10 where values of 0.5 and higher give a score value of 1 and lower values a score of 0. This way the contributor can respond with terms new to the image and still receive a positive score. If the player responds with the most frequent term for the image this image is shown on the right side of the screen. Below the last top answer image empathy shows some statistics about the player. The game does not use quality management.

The second game *GuessIt* is build based on *Empathy* but gives more feedback to a player using *GameLab*. Players can earn badges for doing a certain amount of requests, submitting words that are not already in the database, not using swear words, etc. When a player responded with a term that is also the most frequent term (top answer) to the shown image the game will add this image to a list. This list is shown to the player at the right side of the game screen. To estimate the quality of responses *GuessIt* uses *GameLab*. After the player submitted her response and the game received the feedback object from the *GameLab* server *GuessIt* shows a feedback screen. This screen reports various values from the feedback object such as use of swear words, spelling errors, and the auto correction if available. *GuessIt* will report the score for the entered term as well as the most similar term for this image based on semantic similarity. *GameLab* can handle various tasks with responses given in natural language. We deliberately chose a known task: labeling images to illustrate the general idea.

```
Input:
c:   number of responses from contributor
i:   true if response found in wordnet
t:   true if response has spelling errors
w:   true if response contains swear words
ws:  count swears in the last 10 responses
s:   similarity to most similar response
st:  trust of most similar response
h:   mean trust of last 10 responses
o:   observed highest term frequency
e:   expected highest term frequency
Output:
trust: estimated quality of the response

d = o != 0 ? e/o : 0.0;

if (!i)           { trust = 0; }
else if (w)       { trust = 0; }
else if (fixation){ trust = -1; }
else if (s>0.75)  { trust = s * st; }
else if (s>=0.4)  { trust = (s * st + d)/2; }
else if (ws)      { trust = 0.0; }
else if (t)       { trust = 0.2; }
else              { trust = h; }
if(trust==0 && !w && !t && c<10)
  trust = (10-c)/10;
```

Results

The first question we want to answer is whether feedback can influence response quality. The main difference between *Emapthy* and *GuessIt* is the feedback a player receives. *GuessIt* explicitly points out if a player acts suspiciously, for instance using swear words or the same word over and over again. To analyze the response quality we hand labeled 500 responses from both games to be either acceptable or not. The response quality of *Empathy* is low. Only 69.6% of the responses are acceptable. The responses include swear words, slang, as well as other undesired artifacts. Furthermore players repeatedly argued about the scoring. The response quality of *GuessIt* is higher. Only 6 out of a random sample of 500 responses were not acceptable! Three of the unacceptable responses were swear words. These responses can be filtered before aggregating final results. Only 3 unacceptable responses could not be filtered. This gives a mean response quality for the unfiltered responses of .988 and .994 for the filtered responses. The previously described behavior was not found in the *GuessIt* results. Player that started to respond with swear words did so only a few times. They either stopped playing or reverted to give acceptable responses. Additionally far less player argued about the scoring mechanism. *GuessIt* in contrast to Empathy uses the similarity metric provided by *GameLab* to score responses. These results show the positive effect of the feedback generated with *GameLab* in our experimental setup. Similar effects have also been reported by Wooten and Ulrich (2011).

References

Miller GA (1995) WordNet: a lexical database for English. Communications of the ACM 38:39–41

Wooten J, Ulrich K (2011) Idea generation and the role of feedback: Evidence from field experiments with innovation tournaments. Available at SSRN 1838733

Wu Z, Palmer M (1994) Verbs semantics and lexical selection In Proceedings of the 32nd annual meeting on Association for Computational Linguistics Stroudsburg, PA, USA, p. 133–138.