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

Using Crowdsourcing to Generate an Evaluation Dataset for Name Matching Technologies*

Alya Asarina and **Olga Simek**
MIT Lincoln Laboratory, Lexington, MA
{*alya.asarina, osimek*}*@ll.mit.edu*

1 Introduction

Crowdsourcing can be a fast, flexible and cost-effective approach to obtaining data for training and evaluating machine learning algorithms. In this paper, we discuss a novel crowdsourcing application: creating a dataset for evaluating name matchers. Name matching is the challenging and subjective task of identifying which names refer to the same person; it is crucial for effective entity disambiguation and search. We have developed an effective question interface and work quality analysis algorithm for our task, which can be applied to other ranking tasks (e.g. search result ranking, recommendation system evaluation, etc.). We have demonstrated that our crowdsourced dataset can successfully be used to evaluate automatic name-matching algorithms.

2 Experiment and Results

2.1 Problem Description and Experiment Design

Our goal was to develop a process for automatically generating an evaluation set for Arabic name matching technologies. We chose to frame this as a *ranking* task, where a small number of potential matches associated with each query name must be ordered from best to worst. The initial set was generated by using Freebase (http://www.freebase.com) to obtain query names, and scraping Google and Bing search results to obtain potential matches for those names. Our experiments (summarized in Table 1) were conducted as Human Intelligence Tasks (HITs) posted on Amazon Mechanical Turk (MTurk). Hand-truthed gold standard data was used for evaluation.

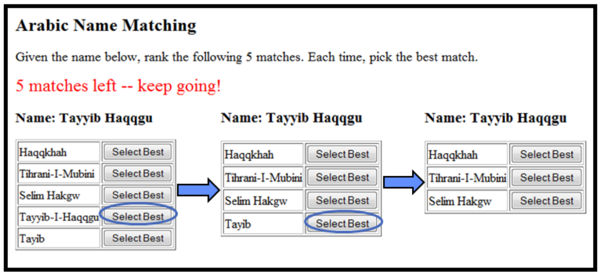
	Rank 5 Exp.	Rank 10 Exp.
# names (= #HITs)	100	100
# matches/name	**5**	**10**
# workers/HIT	30	30

Table 1: Summary of Experiments

2.2 Interface Design

We evaluated a number of ranking interface designs making use of radio button matrices, drag-and-drop lists, select buttons, and type-in answer boxes on a small data set. The most successful design for ranking 5 matches per name (the Rank 5 Experiment) is shown in Figure 1. In this interface, workers iteratively click on the best match, which subsequently disappears.

For ranking 10 matches per name (the Rank 10 Experiment), we used a two-step process inspired by (Marcus et al. 2011). First, workers evaluated matches on an absolute scale using the radio button interface shown in Figure 2. Second, workers were asked to use a drag-and-drop interface to correct the preliminary aggregated rankings derived from the match scores obtained in step 1. For 10 matches, this design significantly improved results.



Arabic Name Matching

Given the name below, rank the following 5 matches. Each time, pick the best match.

5 matches left -- keep going!

Name: Tayyib Haqqgu

Haqqkhah	Select Best
Tihrani-I-Mubini	Select Best
Selim Hakgw	Select Best
Tayyib-I-Haqqgu	Select Best
Tayib	Select Best

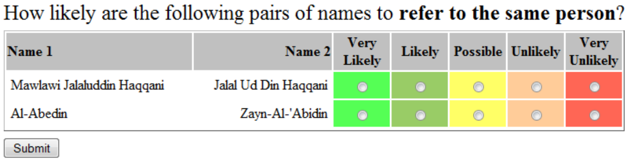
Name: Tayyib Haqqgu

Haqqkhah	Select Best
Tihrani-I-Mubini	Select Best
Selim Hakgw	Select Best
Tayib	Select Best

Name: Tayyib Haqqgu

Haqqkhah	Select Best
Tihrani-I-Mubini	Select Best
Selim Hakgw	Select Best

Figure 1: Design for the Rank 5 Experiment



How likely are the following pairs of names to **refer to the same person**?

Name 1	Name 2	Very Likely	Likely	Possible	Unlikely	Very Unlikely
Mawlawi Jalaluddin Haqqani	Jalal Ud Din Haqqani	○	○	○	○	○
Al-Abedin	Zayn-Al-'Abidin	○	○	○	○	○

Figure 2: Design for the Rank 10 Experiment, Step 1

2.3 Result Analysis

In this section, we discuss the results of our MTurk ranking experiments. Incorporating work quality analysis into the response aggregation process yielded notable quality improvement in both experiments. In particular, a novel work

*This work is sponsored by the Department of Defense under Air Force Contract #FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

quality analysis algorithm designed specifically for ranked data was found to be most effective for our dataset.

We compared three approaches to work quality analysis and evaluated them by comparing the results with 100-name gold standard datasets.

- **Agreement with Plurality:** As a simple baseline approach, we evaluated workers based on their agreement with the ranks assigned to matches by plurality vote. We chose $|i-j|$ as the penalty for giving a match rank i when the plurality choice is j. We used the worker quality estimates obtained to discard low-quality workers and reweight the remaining votes based on worker quality.
- **Get Another Label (Ipeirotis, Provost, and Wang 2010):** The ranking nature of our task was incorporated into the algorithm by assigning different weights to different types of errors.
- **EM Algorithm for Ranking:** We designed our own work quality algorithm, which follows the same approach as (Dawid and Skene 1979; Ipeirotis, Provost, and Wang 2010) but is specifically adapted to ranking tasks. Each worker's quality is evaluated based on the Spearman rank correlation between that worker's responses and the current estimated rankings. Worker qualities are used to reweight the votes and iteratively re-estimate the correct rankings.

	Rank 5 Exp.	Rank 10 Exp.
No quality analysis	.838	.832
Agreement with Plurality	.872	.853
Get Another Label	.861	.496
EM Algorithm for Ranking	**.876**	**.861**

Table 2: Work Quality Evaluation (average Spearman correlation with gold standard)

As seen in Table 2, the *EM Algorithm for Ranking* yielded the best results on this task. The improvement over the next best method in the Rank 10 Experiment is statistically significant. The histogram in Figure 3 shows the accuracy of individual workers in the Rank 5 Experiment. As can be seen, aggregate accuracy is higher than the accuracy of any individual worker.[1]

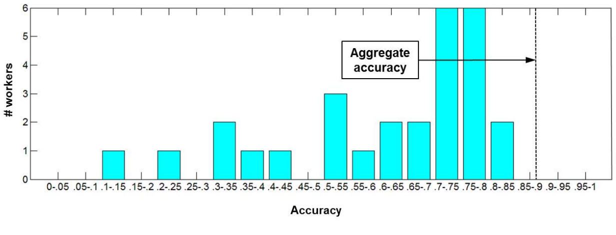


Figure 3: Worker Accuracy Histogram for the Rank 5 Experiment (workers with 10+ HITs)

3 Name Matcher Evaluation

To determine whether the data obtained from MTurk is accurate enough to correctly rank name matchers, we selected four matchers for evaluation (see Table 3). We evaluated the matchers using our 100 name gold-standard dataset for the Rank 10 Experiment, and compared the results to evaluation using the full 1,500 name MTurk dataset we collected. As shown in Table 3, the ranking of matchers by average Spearman correlation based on MTurk data is the same as the ranking based on gold standard data, which indicates that a crowdsourced dataset can be successfully used for name matcher evaluation. Moreover, the large MTurk dataset allows matchers to be ranked with a very high degree of confidence ($p < .000005$ for all pairwise differences).

Rank	Matcher	vs. Gold	vs. MTurk
1.	LingPipe	.785	.792
2.	Basis	.758	.763
3.	MIT LL	.619	.649
4.	Levenshtein	.544	.599

Table 3: Evaluating Name Matchers (average Spearman correlation)

4 Conclusions & Future Work

We have successfully used crowdsourcing to generate a data set for evaluating name matchers, a technology with applications ranging from customer relationship management to regulatory compliance to homeland security.

More generally, we have explored how crowdsourcing can be used to obtain accurate ranking data even when the task is challenging and subjective. Potential application domains include ranking search results by relevance or ranking other workers' responses as a component in a crowdsourcing workflow, among many others. We have proposed effective designs for ranking a small number of options (the Rank 5 Experiment design), or a larger number (the Rank 10 Experiment design). We also described a novel EM algorithm for worker quality analysis that is specifically designed for ranking tasks. Future research is needed to identify the full range of problems where our contributions can be applied. What task parameters determine design effectiveness? How robust is our worker quality algorithm to various parameters, such as the number of choices being ranked, or the distribution of worker qualities (e.g. Gaussian or bimodal)?

References

Dawid, A., and Skene, A. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics* 20–28.

Ipeirotis, P.; Provost, F.; and Wang, J. 2010. Quality management on Amazon Mechanical Turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, 64–67. ACM.

Marcus, A.; Wu, E.; Karger, D.; Madden, S.; and Miller, R. 2011. Human-powered sorts and joins. *Proceedings of the VLDB Endowment* 5(1):13–24.

[1] Similarly, aggregate accuracy for the Rank 10 Experiment is far above average worker accuracy.