Utilization of Synergetic Human-Machine Clouds: A Big Data Cleaning Case

Deniz Iren
Middle East Technical University
Informatics Institute
Ankara, Turkey
+90 312 2103394
diren@metu.edu.tr

Gokhan Kul
Middle East Technical University
Computer Center
Ankara, Turkey
+90 312 2103370
gkul@metu.edu.tr

Semih Bilgen
Middle East Technical University
Department of Electrical and Electronics Engineering
Ankara, Turkey
bilgen@metu.edu.tr

ABSTRACT
Cloud computing and crowdsourcing are growing trends in IT. Combining the strengths of both machine and human clouds within a hybrid design enables us to overcome certain problems and achieve efficiencies. In this paper we present a case in which we developed a hybrid, throw-away prototype software system to solve a big data cleaning problem in which we corrected and normalized a data set of 53,822 academic publication records. The first step in our solution consists of utilization of external DOI query web services to label the records with matching DOIs. Then we used customized string similarity calculation algorithms based on Levensthein Distance and Jaccard Index to grade the similarity between records. Finally we used crowdsourcing to identify duplicates among the residual record set consisting of similar yet not identical records. We consider this proof of concept to be successful and report that we achieved certain results that we could not have achieved by using either human or machine clouds alone.

Categories and Subject Descriptors
H.3.5 [**Information Storage and Retrieval**]: Online Information Services – *Web-based services.* H.5.3 [**Information Interfaces and Presentation**]: Group and Organizational Interfaces – *Collaborative computing.* I.2.7 [**Artificial Intelligence**]: Natural Language Processing – *Text analysis.* I.5.4 [**Pattern Recognition**]: Applications – *Text processing.*

General Terms
Algorithms, Design, Human Factors.

Keywords
Crowdsourcing, Cloud Computing, Crowdservice.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
CSI-SE'14, May 31 - June 07 2014, Hyderabad, India
Copyright 2014 ACM 978-1-4503-2857-9/14/05 $15.00.
http://dx.doi.org/10.1145/2593728.2593733

1. INTRODUCTION
Part human – part machine entities have been a popular concept in science fiction. Although singularity may still be far, peculiar examples of human-machine cooperation and collaboration recently began emerging. Without doubt certain tasks can be performed better by computers. Nevertheless, many tasks still exist which machines fail to outperform humans. Thus, synergy is possible through such human-machine cooperation. Hybrid designs which combine strengths of the machine - human cloud and overcome weaknesses of each make use of the so called "*global brain*" [3].

In the last decade cloud computing and specialization of software services are growing trends much similar to specialization of labor workers in human societies happened back in 1900s.

With the rise of crowdsourcing, harnessing the wisdom of crowds and utilizing human cognition in a scalable way became possible, feasible and efficient. Crowdsourcing is used as a business model for solving a diverse range of problems. Novel API's provided by major crowdsourcing platforms even make it possible to integrate the capabilities of the crowd with the software.

This study addresses a hybrid big data cleaning and migration solution recently undertaken in Middle East Technical University (METU) in Ankara, Turkey. METU employs approximately 2,500 academic and 3,000 administrative personnel. Along with the students, the number of campus residents who use IT services provided by the Computer Center (CC) rises above 30,000. IT structure combines a large number of legacy applications and contains a huge amount of data. Recently a project was initiated to integrate key components of this IT structure as automated business processes. This major overhaul caused some of the legacy data to be migrated as new systems using these data are developed.

One of the legacy applications is the *CV-Academic* which keeps track of academic accomplishments of researchers, including metadata of publications. However the legacy system was designed to allow users to enter the metadata in a free text format which causes inconsistencies while normalizing databases to be used by the newly developed systems. Initially there were 53,822 records with duplicates, typographical errors and mismatching record entries caused by free text entry.

In this paper we describe a hybrid solution which combines DOI query web services, customized string similarity calculation algorithms and crowdsourcing in order to solve this real-life data cleaning and normalization problem. Our business goal is to

eliminate duplicate entries, clean typographical errors and standardize the record data by matching records with external publication repositories. Our research goal is to provide a proof of concept to show that crowdsourcing can be used effectively and efficiently as part of software engineering practices.

Section 1 makes a brief introduction to the problem and the scope of this research. Section 2 reviews related work in the literature and Section 3 describes the proposed solution. Finally Section 4 presents the conclusion and future work.

2. RELATED WORK

Development of hybrid systems which contain crowdsourcing and cloud computing components are considered as a major driver in contemporary IT [2].

Although various names are used to refer this concept, the idea underlying the hybrid systems is similar. Additional value is created through synergy of human and computational services. By integrating them, providing an augmented service is possible which neither can achieve by itself [6]. This new entity formed by increased connectivity and scale of human-computer networks is referred to as the *global brain* by Bernstein et al. [3]. Lenk et al. introduce the cloud ecosystem as a stack of various service layers and places crowdsourcing as topmost layer which is called *human-as-a-service* [12]. Lackermair describes hybrid cloud architecture for online commerce which also includes human-as-a-service layer [11]. Vukovic introduces a cloud-enabled crowdsourcing service [18]. Bernstein introduces a method to improve internet search experience and search accuracy by automated query log mining combined with paid crowdsourcing [4]. Hybrid solutions are used in solving big data problems [1], knowledge management [7], prediction markets [17], mass collaboration [19], open innovation [9] and scientific problem solving [8].

Nevertheless hybrid solutions are not necessarily always successful. Bernstein et al. emphasize the importance of understanding key characteristics of hybrid systems and what makes developing hybrid systems different than traditional computer systems in order to develop successful hybrid systems [3].

3. A HYBRID SOLUTION

Cleaning 53,822 publication records may not be considered a big data problem but still it is not feasible to tackle manually. Furthermore the rest of the legacy application data will be migrated in the near future. Thus, establishing a proof of concept and letting the upper management acknowledge this hybrid approach as a valid solution is critical for future studies.

As a means of solving this data cleaning problem we followed a hybrid approach consisting of multiple phases. The goal of each phase was to eliminate as many duplicate records as possible, minimizing the number of records left for further analysis in upcoming phases.

3.1 Querying DOI Database

In the first phase we used CrossRef [5] DOI Search web services. We developed a simple application which calls the external web service, querying in batches of 20 records per transaction to optimize header – data payload efficiency. We enabled the *fuzzy search* option so that close enough records were also matched with a DOI. If the query was resolved, the record was tagged with the matching DOI. DOI field was used in record comparison in further phases.

DOI resolution process took 40 hours to complete. As a result of querying CrossRef DOI Search web service, 5,681 (10.56% of all records) records were matched with DOIs, 39,415 were unresolved and 391 records could not be processed due to special characters they contain. 8,335 records were excluded from the query as they belonged to specific publication categories without DOIs.

3.2 String Similarity

We define the equality of publication records as *title*, *authors* and *publisher* metadata fields being identical unless they were assigned with DOIs. DOI field has the comparison priority over other fields because it is the global unique identifier assigned to publications. For those records which were unresolved in DOI query phase, *title*, *authors* and *publisher* fields were used for record comparison. However identifying identical records via string comparison can be inaccurate. The reason is that there may be other records for the same publication that somehow contain differences such as mistyped words and abbreviations. Thus while identifying identical records; we may be reaching to the wrong conclusion that there are no other records which belong to the same publication entity.

Therefore we analyzed the record set to identify similar records. Each record was compared with the rest of the records and a Similarity Score (SS) and Similarity Unique Identifier (SUID) were assigned to each similarity instance. We used Levenshtein Distance (LD) and a variant of Jaccard Index (JI') in combination to calculate the SS.

If both LD an and JI' yields an SS of 1 which indicates records being identical, and there is no other similar record identified, then the records are labelled as identical and removed from the record set.

Similarity groups are formed when SS is above certain threshold for both LD and JI'. The threshold value is highly dependent on the language. We tested the algorithms on a test record set of 50 records with various threshold values. We reached to a conclusion that the most suitable threshold for LD is 0.7 and JI' is 0.5. Test results show that all records belonging to the same publication are identified in the same similarity group. Only 18% of the records identified as similar while actually being different. Identifying different records falsely as similar is an error which is preferred over failing to identify records of the same publication in the same similarity group because the second type of error can be detected in further phases while the first type is finalized.

Record difference is decided upon having only an LD score less than 0.7. Therefore any record without an SS equal to or greater than 0.7 are identified as unique and are removed from the record set.

Upon completion of SS calculation phase, 4,558 records were identified as the same; 38,830 records were identified as unique. These records are normalized and removed from the record set leaving 10,434 records for processing in further phases.

3.2.1 Utilization of LD

Prior to comparison with LD, strings are converted to uppercase and special characters in them are either converted to ASCII

equivalents or removed. The *authors* field is standardized via string operations.

LD is used to compare strings and calculate a distance score representing the number of necessary changes in one string to transform into the other. LD algorithm is used as it is defined in the literature [14, 15].

3.2.2 Utilization of JI'

Upon completion of LD calculations JI' is used to compare the records according to the words they contain. Prior to running the algorithm words with character count less than 3 and words "THE", "FROM" and "FOR" were removed from the string.

JI', a variation of Jaccard Index [13], complements LD in order to avoid inaccurate decisions regarding the strings containing same words in different order.

The difference between Jaccard Index and JI' is shown in the (1) and (2).

(1) Jaccard Index = $A \cap B / A \cup B$

(2) JI' = $A \cap B / A$, where $|A| \geq |B|$

This alteration in the algorithm adjusts the accuracy of identification of same and different records. This case dependent alteration eliminates a frequent error in which users enter *title*, *publisher* and/or *year* concatenated into the *title* field. Jaccard Index calculates a lower similarity score compared to JI', due to increased number of words in $A \cup B$. In this case JI' provides better accuracy compared to the original Jaccard Index.

3.3 Crowdsourcing

The remaining 10,434 records, which both specialized web services and the algorithms failed to classify, were left for the crowdsourcing phase. In this phase we aimed at harvesting the ability of human mind to recognize the differences in similar text fields.

These 10,434 records are separated into 4,359 groups according to their SUIDs. The numbers of publications in similarity groups of varying sizes are shown in Table 1.

Table 1. Numbers of publications in similarity groups

Group Size	Record Count	Group Size	Record Count
15	1	8	9
14	0	7	14
13	1	6	32
12	1	5	53
11	3	4	248
10	6	3	637
9	9	2	3345

Since the design of crowdsourcing tasks strongly affects the task performance, we rearranged the data into pairs. Therefore we were able to ask the simple question with limited binary answers: "*Is the following record pair the same or different?*" The pairing task caused increase in the number of tasks to be crowdsourced. The formula for calculating the number of pairs in a group is:

$$\# \text{ Pairs} = \text{Group Size} \cdot (\text{Group Size} - 1) / 2$$

Thus the total number of tasks to be crowdsourced is calculated as 9,308.

These tasks were published on Amazon Mechanical Turk (AMT). Workers were asked to evaluate 4 pairs of records in each task. Upon successful completion they were paid 0.02$. 1 of 4 pairs was selected from a gold standard pair set and the success of the task was decided according to the result submitted for the gold standard pair.

Due to loose employee-worker relationship, anonymity and diverse skills of the crowd, crowdsourcing leads to poor quality in end product. Therefore crowdsourcing practitioners use certain quality assurance mechanisms [10].

In this study we used multiple quality assurance mechanisms in conjunction, including *gold standard*, *redundancy*, and *automatic check* [10, 16].

First level of quality control was performed using gold standard microtasks. Prior to the crowdsourcing phase we formed a gold standard record set consisting of 100 record pairs. 50 record pairs in gold standard record set were positive examples which can easily be identified as the same and the remaining 50 record pairs were negative examples which are unmistakably different. Each task contained 1 gold standard and 3 regular record pairs. Those of which submit correct results for the gold standard task were accepted.

We assigned each task to three different users redundantly. Later on we used majority decision technique to derive a final decision regarding the publication records.

Finally we automatically checked if the decisions regarding the record pairs are consistently transitive in their respective similarity groups. A few inconsistencies were identified and resolved automatically by considering the most frequent decision as true.

Crowdsourcing phase was completed in 17 days and cost $186. 9,308 microtasks were performed by 1,385 workers and 27,924 decisions were collected. 1,920 decisions were not accepted due to gold standard task failure. The average number of tasks completed in one day is 1,643. Average time to complete one microtask is 52 seconds.

6,224 record pairs were identified as same and 3,084 different. These decisions were used to automatically derive final judgments whether or not records are same within each similarity group.

We used expert evaluation with random sampling for judging the accuracy of crowdsourcing. We performed 1,500 random microtasks manually to derive an expert evaluation set as a basis for comparison. Upon completion of expert evaluation we observed that 96 of the decisions did not match with the expert evaluation displaying an error rate of 6.4% of all records processed in crowdsourcing phase.

4. CONCLUSION & FUTURE WORK

In this study we developed a prototype software solution for a data cleaning problem. This hybrid solution included DOI search

web services, customized string similarity calculation algorithms and microtask crowdsourcing.

Compared to assigning this type of task to an employee, crowdsourcing yields significant cost and time savings. Furthermore, we observed repetitive overdoing the same microtask can be very boring and psychologically challenging. Thus, crowdsourcing the job is beneficial so that many people can work on microtasks and share the psychological burden.

Crowdsourcing is imperfect by design. Results of mediocre quality are easily achievable. The overall error rate we observed in the crowdsourcing phase of this study was 6.4% (596 of 9,308), which is acceptable for this particular job. Better quality assurance designs are needed in order to increase the accuracy level. Upon completion of the crowdsourcing phase residual errors (596 of 53,822) were introduced in the newly developed system which includes functionality for the authors to manually correct the records. The authors were informed about the data migration process and are encouraged to review their publication records and fix the problems they detect.

Beside effectiveness, managing the cost of quality of crowdsourcing is also important. We propose utilization of certain cost models to estimate quality costs and select cost optimized quality assurance mechanisms.

As a result, we consider using such hybrid approaches in certain problems feasible. Combining computational power of computers with human skills and cognition allows us to leverage the strengths while minimizing the weaknesses of both. We acknowledge crowdsourcing as a valuable method of problem solving not only in software development and data analysis but also in a wide variety of research areas of the university.

Our main contribution consists of the outcomes of this case, in which we utilized crowdsourcing as part of prototype software solution for data cleaning. We present our experiences in detail which may guide other researchers and practitioners facing problems that can benefit from utilizing crowdsourcing. Our observations regarding the quality may provide useful insight, aiding practitioners to make realistic expectations while encouraging them to see such hybrid approaches as a valid way of problem solving in software engineering.

Furthermore, our secondary contribution is the customized JI' algorithm used in combination with LD and threshold values we observed to yield accurate string similarity calculation, which may be directly used in solving similar data cleaning problems.

Our research will continue with two interrelated focuses: Developing effective and efficient ways of integrating crowdsourcing within big data analysis process and guidelines for selection and cost estimation of crowdsourcing quality assurance mechanisms.

5. ACKNOWLEDGMENTS

This research was funded by METU Research Project Funding Office (METU BAP) and conducted by Integrated Information Systems (IIS) project personnel of METU Computer Center. We thank our project teammates for their innovative ideas, positive feedback and continuous effort to support this research.

6. REFERENCES

[1] Von Ahn, L. and Dabbish, L. 2004. Labeling images with a computer game. *Proceedings of the 2004 conference on Human factors in computing systems - CHI '04*. (2004), 319–326.

[2] Amer-Yahia, S., Doan, A., Kleinberg, J., Koudas, N. and Franklin, M. 2010. Crowds, Clouds, and Algorithms: Exploring the Human Side of "Big Data" Applications. *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data* (New York, NY, USA, 2010), 1259–1260.

[3] Bernstein, A., Klein, M. and Malone, T.W. 2012. Programming the Global Brain. *Commun. ACM*. 55, 5 (May 2012), 41–43.

[4] Bernstein, M.S., Teevan, J., Dumais, S., Liebling, D. and Horvitz, E. 2012. Direct Answers for Search Queries in the Long Tail. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2012), 237–246.

[5] CrossRef: *www.crossref.org*.

[6] Davis, J.G. 2011. From Crowdsourcing to Crowdservicing. *Internet Computing, IEEE*. 15, 3 (May 2011), 92–94.

[7] Fast, E., Steffee, D., Wang, L., Brandt, J. and Bernstein, M.S. 2014. Emergent, Crowd-scale Programming Practice in the IDE. (2014).

[8] Fold-it: *fold.it*.

[9] Innocentive: *www.innocentive.com*.

[10] Iren, D. and Bilgen, S. 2013. *Cost models of crowdsourcing quality assurance mechanisms*.

[11] Lackermair, G. 2011. Hybrid cloud architectures for the online commerce. *Procedia Computer Science*. 3, 0 (2011), 550–555.

[12] Lenk, A., Klems, M., Nimis, J., Tai, S. and Sandholm, T. 2009. What's Inside the Cloud? An Architectural Map of the Cloud Landscape. *Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing* (Washington, DC, USA, 2009), 23–31.

[13] Levandowsky, M. and Winter, D. 1971. Distance between Sets. *Nature*. 234, 5323 (Nov. 1971), 34–35.

[14] Levenshtein, V. 1965. Binary codes capable of correcting spurious insertions and deletions of ones. *Problems of Information Transmission*. 1, (1965), 8–17.

[15] Navarro, G. 2001. A Guided Tour to Approximate String Matching. *ACM Comput. Surv.* 33, 1 (2001), 31–88.

[16] Quinn, A.J. and Bederson, B.B. 2011. Human Computation: A Survey and Taxonomy of a Growing Field. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2011), 1403–1412.

[17] Tziralis, G. and Tatsiopoulos, I. 2007. Prediction Markets: An Extended Literature Review. *Journal of Prediction Markets*. 1, 1 (2007), 75–91.

[18] Vukovic, M. 2009. Crowdsourcing for Enterprises. *Proceedings of the 2009 Congress on Services - I* (Washington, DC, USA, 2009), 686–692.

[19] Wikipedia: *www.wikipedia.org*.