CROWCFIL: A FRAMEWORK FOR CONTENT FILTERING IN CROWDSOURCING ENVIRONMENT

O. O. Bamgboye, A. A. Orunsolu, M. A. Alaran, A. A. Adebayo, M. A. Oyeleye

Department of Computer Science, Moshood Abiola Polytechnic, South West, Nigeria

Corresponding author: O.O. Bamgboye, bamgboye.oluseun@mapoly.edu.ng

ABSTRACT: The growth of internet connectivity and bandwidth has now made it possible to harness "human computation" in near-real time from a vast and ever-growing, distributed population of online internet users. In the process of distributing and managing knowledge online, so many concepts arose in which crowdsourcing is an example that cannot be overlooked. Crowdsourcing depends on human worker but human worker are prone to errors. To leverage the power of crowdsourcing, in this paper, a framework called Crowdsourcing Content Filtering (CrowCFil) System was designed. CrowCFil is a framework designed to exploit the conventional crowdsourcing techniques in order to improve the reliability and integrity of information given by contributors to requesters on a crowdsouring platform. It consists of three major functional modules: Task Initiator Module, Contributor Module and CrowCFil Engine Module, all of which are interdependent. The core part of System is the CrowCFilS Engine Module, which gives the system the power to check for the reliability and integrity of response as submitted by a contributor with the aid well defined algorithm embedded into a set of interrelated functions present in it. The framework is suitable for implementation in a relatively large distributed crowdsourcing platform while keeping the cost of operating a crowdsourcing low.

KEYWORDS: Crowdsourcing, knowledge management, Contributor, Requester, Filtering System.

1. INTRODUCTION

Knowledge Management (KM) is a discipline that seeks to improve the performance of individuals and organizations by maintaining and leveraging the present and future value of knowledge assets. Knowledge management systems encompass both human and automated activities and their associated artifacts. As an interdisciplinary discipline, KM regroups concepts from Information Technology Management, Philosophy, Cognitive Sciences, and Organization Studies. Knowledge Management has caused a shift from a transaction to a Distributed Knowledge Management (DKM) perspective on inter-organizational information processing. The DKM concept structures the knowledge creation, knowledge sharing and knowledge exploitation in organizations according to a product state model (PSM) required for management of technological diversity. Each player in the network acquires specific knowledge from other players for decision support.

Crowdsourcing as one the major application area of Distributed Knowledge Management (DKM) is the process of obtaining needed services, ideas, or content by soliciting contributions from a large group of people, and especially from an online community, rather than from traditional employees or suppliers.

The rapid development of Web 2.0 and social media has greatly contributed to the fast rising evolution of crowdsourcing. The Internet provides a particularly good venue for crowdsourcing since individuals tend to be more open in web-based projects where they are not being physically judged or scrutinized and thus can feel more comfortable sharing. This ultimately allows for well-designed artistic projects because individuals are less conscious, or maybe even less aware, of scrutiny towards their work.

Crowdsourcing has become a powerful mechanism for outsourcing tasks which seems like a great solution; put your problem out there, and wait for a solution from the masses which includes both experienced and inexperienced contributors. Many organisations have been engaging in crowdsourcing as a way of finding solutions to difficult technical problems. Some of these organization engage directly with the crowd, while others have used intermediaries or 'expert networks' who run crowdsourcing platforms as a product.

However, humans are more effective than computers for many tasks, such as identifying concepts in images, translating natural language, and evaluating the usefulness of products. Thus, there has been a lot of recent interest in crowdsourcing, here humans perform tasks for pay or for fun.

The rest of the paper is organized as follows; Section 2 contains the review of related works in the field of crowdsourcing, Section 3 presents the different modules for the architecture of the content filtering system while Section 4 presents the conclusion and future work.

2. RELATED WORKS

John Le et al. ([J+10]) presented a paper which assessed how the dynamic learning environment can affect the workers' results in a search relevance evaluation task completed on Amazon Mechanical Turk. The paper showed how the distribution of training set answers impacts training of workers and aggregate quality of worker results. It was concluded that in a relevance categorization task, a uniform distribution of labels across training data labels produces optimal peaks in 1) individual worker precision and 2) majority voting aggregate.

Crowdsourcing has become a powerful mechanism for outsourcing tasks, which are traditionally performed by a specialist or small group of experts, to a large group of humans ([Gre11]). It is used for a variety of applications, such as evaluating ideas, creating knowledge repositories, or developing new products collaboratively.

The research of crowdsourcing is a vigorous research area that has been steadily increasing over the last several years ([ZZ12]) and there is still an ongoing need for scientific engagement in this field ([HH12; L+09]). Crowdsourcing has been used for a variety of applications, such as evaluating ideas, creating knowledge repositories, or developing new products collaboratively.

In the view of [DDC12], Crowdsourcing is becoming a valuable method for companies and researchers to complete scores of micro-tasks by means of open calls on dedicated online platforms. Crowdsourcing results remains unreliable, however, as those platforms neither convey much information about the workers' identity nor do they ensure the quality of the work done. Instead, it is the responsibility of the requester to filter out bad workers, poorly accomplished tasks, and to aggregate worker results in order to obtain a final outcome". The work reviewed techniques currently used to detect spammers and malicious workers, whether they are bots or humans randomly or semi-randomly completing tasks; then, they described the limitations of existing techniques by proposing approaches that individuals, or groups of individuals, could use to attack a task on existing crowdsourcing platforms. Focus was also laid on crowdsourcing relevance judgments for search results as a concrete application of our techniques.

[L+12] implemented a Crowd sourcing Data Analytics System, CDAS. A framework was designed to support the deployment of various crowd sourcing applications. The core part of CDAS is a quality-sensitive answering model, which guides the crowd sourcing engine the power to process and monitor the human tasks. To show the effectiveness of the model, it was implemented and deployed on two analytics jobs, a twitter sentiment analytics job and an image tagging job which used real Twitter and Flickr data as queries respectively. The approaches were then compared with state-of-the-art classification and image annotation techniques. The result showed that by embedding the quality sensitive model into crowd sourcing query engine, it will effectively reduce the processing cost while maintaining the required query answer quality.

[P+14] presented a research aimed at improving the learning experience of existing how-to videos with step-by-step annotations by designing a workflow which does not rely on domain-specific customization, works on top of existing videos, and recruits untrained crowd workers. The author first performed a formative study to verify that annotations are actually useful to learners before creating ToolScape, an interactive video player that displays step descriptions and intermediate result thumbnails in the video timeline which helps the learner to perform better and gained more self-efficacy than the traditional video player. A novel crowdsourcing workflow was then introduced to add the needed step annotations to existing how-to videos at scale which extracts step-by-step structure from an existing video, including step times, descriptions, and before and after images. A Find-Verify-Expand design pattern was introduced for temporal and visual annotation, which applies clustering, text processing, and visual analysis algorithms to merge crowd output. The workflow was evaluated with Mechanical Turk, using 75 cooking, makeup, and Photoshop videos on YouTube. It was then concluded that the workflow can extract steps with a quality comparable to that of trained annotators across all three domains with 77% precision and 81% recall.

A survey by Balan ([BP14]) reviewed many areas where the problem of string similarity matching search appears and one of the most demanding is information retrieval to find relevant information in text collection. The author then surveyed and presented an overview of string similarity matching and also comparison of different algorithms to conclude the better performance on searching the text and the important tool is named as string matching.

3. ARCHITECTURE OF CROWCFILS

In Figure 1, the complete architecture of the content filtering system is presented. The Filtering System for Crowdsourcing is a system that exploits the crowdsourcing techniques to improve the reliability and integrity of information on a crowdsourcing platform. The core difference between System and every other conventional crowdsourcing systems lies

in the ability of the system to filter only Information with high similarity index that is provided by the knowledge worker. The System relies solely on enhanced filtering algorithm and query processing technique where other crowdsourcing systems employ human workers to assist in the analyzing of tasks produced by knowledge worker. The architecture of CrowCFil system consists of three major functional modules namely; Task Initiator Module, Contributor Module and CrowCFil Engine Module.

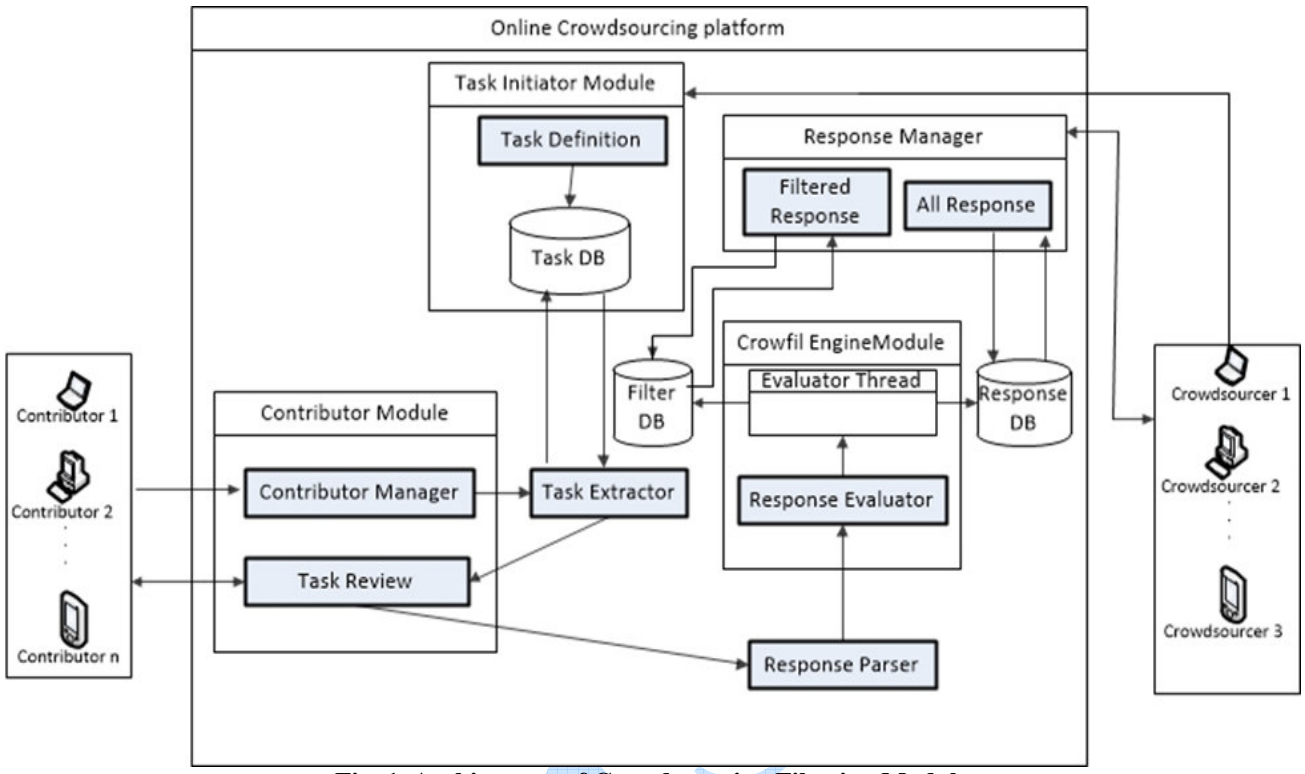

Fig. 1. Architecture of Crowdsourcing Filtering Module

3.1 Task Initiator Module

The Task Initiator Module receives task from the crowdsourcer which is fed to the System. The crowdsourcer is the individual imposed with the responsibility of defining the task to done by respective knowledge worker. The Task Initiator Module defines how task is to be structured which is often initiated through the Task Definition. The Task Definition uses the Task Manager to provide a way to assist the crowdsourcer to be able to define individual task. Compared to other crowdsourcing system, the Task Definition also learns the crowdsourcer experience and expectations about the task by allowing the crowdsourcer define various keywords around the task. Once this is established, the Task Definition automatically stores and updates the task pattern in the TaskDB for future use.

3.2 Contributor Module

The concurrent usage of the internet connection to access task through CrowCFil system provides mechanism to manage several knowledge workers on the same task concurrently through the use of Contributor Module. The Contributor Module is made up of two distinct parts namely; Contributor Manager and Task/Response Manager.

3.2.1 Contributor Manager

This component manages several contributors currently on the CrowCFil system. It provides HTTPRequest and HTTPresponse which are set of protocols used to initiate request and receives response respectively by the CrowCFil system. Each knowledge worker uses HTTPRequest to make a request for available tasks on CrowCFil system and the request is automatically sent to the Task/Response Manager for further processing.

3.2.2 Task/Response Manager

The request sent by the Contributor Manager is received by the Task Extractor. The Task Extractor sends request and receives response from the TaskDB respectively. The response received is then transferred to Task Review. The Task Review through its View Module sends the available tasks to that specific knowledge worker that initiated the request from the Contributor Manager using the HTTPResponse. Once the knowledge worker have reviewed and provided an insight to the task available, the response is then accepted by the Response Submission Module. The response is later sent to an interface called Response Parser for further processing.

3.3 CrowCFil Engine Module

This Module is the key aspect of the Filtering System for Crowdsourcing. It checks for the reliability and validity of each response submitted by each knowledge worker with the use of set of standard procedures embedded within it. At the initial stage, the output of Task/Response Manager is received by the Response Parser. The Response Parser acts as an interface between the Contributor Module and the CrowCFil Engine. It tranfers the response from the Task/Response Manager in the Contributor Module to the CrowCFil Engine. The Response Evaluator receives the response from the Response Parser and automatically initializes a synchronous thread called Evaluator Thread (ET). The ET concurrently handle knowledge worker's response through an inbuilt-procedure. The Evaluator Thread consists of two interdependent functions and one procedure; RemoveStopword, CFilter and RateResponse respectively.

3.3.1 RemoveStopword

The response is passed into the RemoveStopword function indicated. The RemoveStopword function uses Rabin-Karp algorithm to break the responses into tokens and removes the stopwords (comprising remaining words apart from the defined keywords) being presented and then returns the remaining words to another function.

3.3.2 CFilter

The output of the function is passed into the CFilter for the purpose of natural language processing and ranking. An offline dictionary is embedded into this function for word similarity check. The filtering of a particular response may be found to be synonymous with defined crowdsourcer's keyword, it is then automatically stored in the TaskDB for the purpose of optimizing the speed of the filtering process. The CFiler checks for the number of the keyword that matches and rates the response based on the match found. After a complete filtering the number of match found is returned by the function.

3.3.3 RateResponse

This is the last in the Evaluator Thread; it is mainly designed for storing response into the appropriate DB. For a match found in the function above, it then checks for the value; if the value is 0 then the response is stored into the ResponseDB alone but otherwise, the response is sent to the FiltredDB and ResponseDB for futher processing by the crowdsourcer.

4. CONCLUSION AND FUTURE WORK

In this work, we have proposed a CrowcFil, an engine that filters and validates the contributions of knowledge workers. By enforcing the keyword specification the system interprets and retrieves the result of search queries with high similarity index. The modules in the system guides the CrowCFil to generate proper response plans for the Crowdsourcer based on the ranking model. The system consists three major functional modules that assists in achieving a measure of reliability of contributions by knowledge workers. The framework provides a robust approach towards the delivery of structured answers or responses to unstructured queries especially with multiple interpretations. Thus, CrowcFil presents a promising approach towards building a crowdsourcing platform that understands user queries and respond with reliable and well validated information.

In future we hope to build an efficient template for the implementation of crowdsourcing platform that will in turn improve the use of the crowd or knowledge worker.

REFERENCES

[Bra08a] **Daren C. Brabham** - *Crowdsourcing as a Model for Problem Solving: An Introduction and Cases*, Convergence: The International Journal of Research into New Media Technologies: 75–90, 2008, archived from the original on April 4, 2012.

[Bra08b] **Daren C. Brabham** - *Moving the Crowd at iStockphoto: The Composition of the Crowd and Motivations for Participation in a Crowdsourcing Application*, 2008.

[Bra10] **Daren C. Brabham** - *Moving the Crowd at Threadless: Motivations for Participation*. in "Managing Unexpected Publics Online: The Challenge of Targeting Specific Groups with the Wide-Reaching Tool of the Internet", International Journal of Communication. Crowdsourcing Application" Information, Communication & Society13: 1122–1145, 2010.

[Bra12] **Daren C. Brabham** - *Motivations for Participation in a Crowdsourcing Application to Improve Public Engagement in Transit Planning*, Journal of Applied Communication Research: 307–325, 2012.

[BP14] **S. Balan, P. Ponmuthuramalingam** - *A Survey on String Similarity Matching Search Techniques*, International Journal

of Emerging Technologies in Computational and Applied Sciences (IJETCAS); IJETCAS 14-624, Pg 286-288, 2014.

[DDC12] **Djellel Eddine Difallah, Gianluca Demartini, Philippe Cudré-Mauroux** - *Mechanical Cheat: Spamming Schemes and Adversarial Techniques on Crowdsourcing Platforms*, CrowdSearch 2012 workshop at WWW 2012, Lyon, France, 2012.

[EG12] **E. Estellés-Arolas., F. González-Ladrón-de-Guevara** - *Towards an integrated crowdsourcing definition*, Journal of Information Science. 38, 189–200, 2012.

[Gre11] **S. Greengard** - *Following the crowd*, Communications of the ACM. 54, 20–22, 2011.

[How06a] **Jeff Howe** - *The Rise of Crowdsourcing*. http://www.wired.com, 2006, accessed on 03 March, 2015.

[How06b] **Jeff Howe** - *Crowdsourcing: Why the Power of the Crowd is Driving the Future of Business* (PDF), The International Achievement Institute, 2008.

[How06c] **Jeff Howe** - *Crowdsourcing: A Definition*, Crowdsourcing Blog, 2006, Retrieved January 2, 2013.

[HH12] **L. Hammon, H. Hippner** - *Crowdsourcing*, Wirtschaftsinformatik. 54, 165–168, 2012.

[J+10] **L. John, E. Andy, H. Vaughn, B. Lukas** - *Ensuring quality in Crowdsourced Search Relevance Evaluation: The effects of training question distribution,* In proceedings of workshop on Crowdsourcing for Search Evaluation, SIGIR pp 17-20 , 2010.

[KC09] **R. Kazman, H. M. Chen** - *The metropolis model a new logic for development of crowdsourced systems*, Communications of the ACM. 52, 76–84, 2009.

[L+09] **J. M. Leimeister, M. Huber, U. Bretschneider, H. Krcmar** - *Leveraging Crowdsourcing: Activation-Supporting Components for IT-Based Ideas Competition*, Journal of Management Information Systems. 26, 197–224, 2009.

[L+12] **Xuan Liu, Meiyu Lu, Beng Chin Ooi, Yanyan Shen, Sai Wu, Meihui Zhang** - *CDAS: A Crowdsourcing Data Analytics System*, School of Computing, National University of Singapore, Singapore College of Computer Science, Zhejiang University, Hangzhou, P.R. China, (PDF), 2012.

[MLD10] **T. W. Malone, R. Laubacher, C. Dellarocas** - *The collective Intelligence genome*, Engineering Management Review, IEEE. 38, 38–52, 2010.

[PL00] **Mogens Kühn Pedersen, Michael Holm Larsen** - *Distributed Knowledge Management Based on Product State Models - The Case of Decision Support in Health Care Administration*, A Forthcoming in Decision Support Systems, Special issue on Knowledge Management, 2000.

[P+14] **Juho Kim Phu, Nguyen Sarah Weir, Philip J. Guo, Robert C. Miller, Krzysztof Z. Gajos** - *Crowdsourcing Step-by-Step Information Extraction to Enhance Existing How-to Videos*, CHI 2014, April 26–May 1, 2014, Toronto, Ontario, Canada, 2014.

[Ros12] **Dawson Ross** - *Getting Result from Crowds*, http://www.crowdsorcing.org/editorial/a-brief-history-of-crowdsourcing-infographics/12532, 2012, accessed on 03 March, 2015.

[ZZ12] **Y. Zhao, Q. Zhu** - *Evaluation on crowdsourcing research: Current status and future direction*, Ji Information Systems Frontiers. 1–18, 2012.