OPEN WATER

A Software System Proposing the Processing of Crowdsourced Data to Monitor a Flood Event: An A.I. Approach

Arya Tanmay Gupta[1]

[1]*Ramanujan College (University of Delhi)*

ABSTRACT

We present Fics (Fetch information through crowdsourcing), a platform that is ready-to-take posts from social media platforms and infers the water heights referred in them. These posts are expected to come from the citizens who are witnessing a flood event in real time. Fics corrects the spacing in the string, translates the string into corresponding mathematical notations and then finally compute the water heights from the posts. The objective of Fics is to provide such a platform that can be used for the citizens from the data received from them only, without making them use a software which is to be installed on their machines separately. Fics employs Artificial Intelligence to infer the required values (water heights) from the posts. Fics ignores the invalid input strings.

Keywords
crowdsourcing, machine intelligence, mathematical notations, disaster, modelling

1.0 Introduction

According to the statistics of Munich RE (2015), there have been total 21,700 loss events between 1980 and 2014, which cover all disasters worldwide. Systematic disaster data collection and analysis can be used to inform policy decisions to help reduce disaster risks and build resilience. [1]

Crowdsourcing in various forms like data, photographs, and questionnaires have proven to be effective tools in models/validation [5]. One of its uses is to monitor the natural calamities. This technology has a potential of giving a lot of assistance to the victims, also the ones who are involving in rescue activities and future modelling of the related environment.

Efforts in the nascent field of human computation have explored methods for gaining programmatic access to people for solving tasks that computers cannot easily perform without human assistance. Human computation projects include work on crowdsourcing, where sets of people jointly contribute to the solution of problems. [11]

Various works like those which are presented in [4] (Social.Water), [6] (CityCAT), [8] (FloodPatrol), [9] (Mapster) present software systems that collect crowdsourced data. Such software agents were to be installed into the machine, mobile phone or computer, of the user and then data is collected from the citizens, which would be used for modelling purposes.

Social media platforms like Twitter receive an overwhelming amount of situational awareness information. For emergency response, real-time disaster insights are important. Finding actionable and tactical information in real-time poses serious challenges. Effective coordination of human and machine intelligence can improve disaster response efforts. [14]

We have developed a software agent using Python programming language which processes synthetic data similar to be fetched through social media during a flood event. We call it Fics. It infers the water heights of a synthetic flood. It takes synthetic inputs which are potential in a post which is expected to come from a person who is stuck in a flood.

We present our motivation in the Literature Review. Then we explain our methodology that we followed to develop this software, the databases and the functioning, finally concluding and proposing how Fics can be integrated with various software in order to model floods.

2.0 Literature review

Floods are one of the most destructive natural disasters and threaten communities and properties [2]. India and Sri Lanka suffered a loss of INR200 crore billion ($3 billion) because of the floods that occurred in November-December 2015, also, loss of at least 386 lives [1].

Since the evolution of computer science, internet, and environmental simulation there has been a high demand of modelling the disasters. We therefore need such systems which can model a disaster fast enough.

One issue in this context is if no satellite is over the area of study, or otherwise also, modelling using images is more complex for a machine to compute.

We therefore take help of crowdsourcing. Various works have been successfully presented in this context, such as in [4], [6], [8] and [9]. But these works were made to be functional after a flood, such as CityCat [6], or otherwise they are data forms on software or web-pages, such as Social.Water [4], FloodPatrol [8], Mapster [9], CyberFlood[15], this way requesting for data in a particular format.

Our idea is to combine human and machine intelligence to monitor and act more efficiently during a disaster such as a flood.

Various works have been done which employ concepts of machine intelligence to model systems. [11] show how machine learning and inference can be harnessed to leverage the complementary strengths of humans and computational agents to solve crowdsourcing tasks. They have developed a software system called CrowdSynth to model the Galaxy Zoo project [18]. They did this by voting. The test set includes 44350 votes collected for 1000 randomly selected galaxies, and 453 SDSS (Sloan Digital Sky Survey) [19] image features describing each galaxy.

[12] have proposed Qurk, a query system for managing the workflows designed of various kinds of HITs (Human Intelligence Tasks) for filtering, aggregating, sorting, and joining data sources together is common, and comes with a set of challenges in optimizing the cost per HIT, the overall time to task completion, and the accuracy of MTurk (Amazon's Mechanical Turk) [17] results. This was to allow crowd-powered processing of relational databases.

[13] had proposed TURKONTROL, a theoretical model based on POMDPs (Partially observable Markov decision process) to optimize iterative, crowdsourced

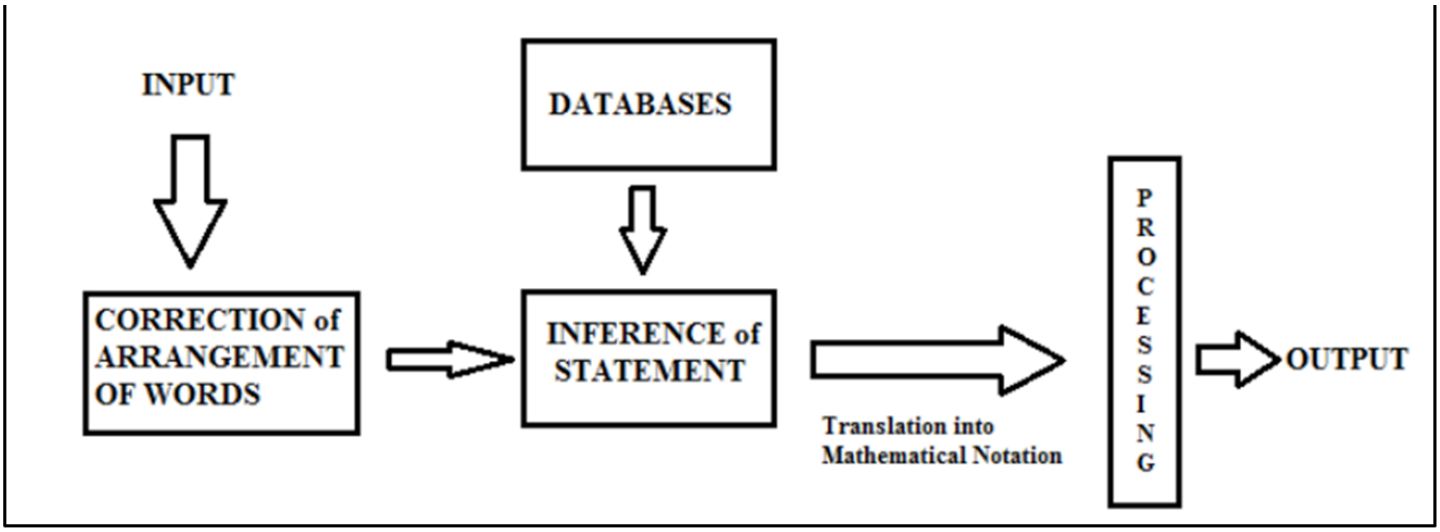

Figure 1. Overview of Fics.

workflows. They have then presented an end-to-end system that first learns TURKONTROL's POMDP parameters from real Mechanical Turk data, and then applies the model to dynamically optimize live tasks.

[14] have presented AIDR (Artificial Intelligence for Disaster Response) to perform automatic classification of crisis-related microblog communications. The objective of AIDR is to classify messages that people post during disasters into a set of user-defined categories of information (e.g., \needs", \damage", etc.) The platform combines human and machine intelligence to obtain labels of a subset of messages and trains an automatic classifier to classify further posts.

[15] has presented CyberFlood, a public cloud-based flood cyber-infrastructure. It collects, organizes, visualizes, and manages several global flood databases in real-time. It provides location-based eventful visualization as well as statistical analysis and graphing capabilities. Crowdsourcing data collection methodology is employed. This system allows modernization of the existing paradigm used to collect, manage, analyze, and visualize water-related disasters.

FDMS [16] (Flood Disaster Management System) has been developed to provide disaster reduction capabilities that cover flood management. This system links environmental models with disaster-related data to support FDMS-constructed workflows with suitable models and recommend appropriate datasets as model input automatically. This system uses OpenMI-based modular design which unifies interfaces and data exchange to provide flexible and extensible architecture. Subsequent 3D visualization improves the interpretability of disaster data and the effectiveness of decision-making processes.

We, in this paper, present an agent which gives independence to the user to use any social media website such as twitter, facebook, google and many others. We infer the posts on these websites and use fundamental concepts of machine intelligence to infer the water heights at various geolocations on earth where probably a flood event has occurred. This information will be further used to model the precise present and potential future of a flood.

3.0 Methodology and results

3.1 Overview

Fics takes a synthetic input, processes, and gives out water height, which can be further used to model a flood. We have to catch the geolocations, tagged with such posts. We have translated a statement in English into corresponding mathematical notations to render values; these values will then be used for computation and modelling. The overview of Fics is given in Figure 1.

3.2 Database

The database of the software includes (1) the height of various objects which can be used in posts for referring heights, (2) list of various helping verbs, (3) symbols, (4) various fundamental and non-fundamental units strings which can be used in the post to refer relative time and height, (5) a list of degrees (such as 'first', 'second', 'third') which can be used in the statements as reference with the objects, say floors of buildings.

3.3 The parent function

The main function passes the string of subject to the function process(). The architecture of process() is as follows.

Process()
Correction of statements
1 SeperateLettersNumbersSymbols(st)
2 st = st.split()
3 TrimArticles()
4 CombineDecimal()
5 HandleAporstrophe()
Inference of water heights
5 CheckHeightUnits()
6 CheckReferredMeasurements()

These are the most important steps represented by the functions of Fics. The working of these functions discussed further in this section.

3.4 Correction of statements

The software takes various steps to correct the string to finally infer the value. Sequentially, they are as follows.

(a) Separate the symbols and numbers from words. A function SeperateLettersNumbersSymbols(st) is defined to do this. The algorithm of this function is as follows.

SeperateLettersNumbersSymbols(st)
1 ***st2*** = ""
2 for ***i*** = 0 to length of ***st***
3 If ***st[i]*** is an alphabet, and the previous letter is a number or symbol

4 ***st2*** = ***st2*** + " " + ***st[i]***
5 Else, If ***st[i]*** is an alphabet, and the previous letter is a number or symbol
6 ***st2*** = ***st2*** + " " + ***st[i]***
7 Else, If st[i] is an symbol, and the previous letter is an alphabet or number
8 ***st2*** = ***st2*** + " " + ***st[i]***
9 Else
10 ***st2*** = ***st2*** + ***st[i]***
11 return ***st2***

(b) Split the string. In this step, the string is split, using the split() function, wherever a character such as ' ' (blank space), '\n' (line break), or '\t' (tab) is encountered. The string thus breaks into an array of words and is stored back in ***st.***

(c) Trim Articles. In this section, the articles of english language are removed from the newly formed array ***st***. A function, TrimArticles() does this task, defined as follows.

TrimArticles(***st***)
1 for ***i*** = 0 to len(***st***)
2 if ***st[i]*** in English.article
3 ***st.***__delitem__(***i***)
4 return st

(d) Detect if a dot (.) is following a value; in that case combine these two and also the string preceding the dot, iff it is a value. This forms the decimal values. Which might have gone undetected, because of the spaces inserted in step a.

A function CombineDecimal() is defined in the program to do this job. The pseudo code of this function is defined as follows.

CombineDecimal(***st***):
1 for ***i*** = 1 to length of ***st***
2 ***word*** = st***[i]***
3 If ***word*** is "."
4 If the string formed by joining ***st[i*** - 1], ***st[i]***, and ***st[i*** + 1] makes a number
5 Join ***st[i*** - 1], ***st[i]***, and ***st[i*** + 1]
8 return ***st***

(e) Handle Apostrophe ('). Convert each string sequence that occurs in the form 'X' '" 's' 'Y' to 'Y' 'of' 'X'. This is important to be handled because the database that we are taking is in the form "Y of X", each time such a potential referred object is used in the statement. The function HandleAporstrophe() is defined as follows.

HandleApostrophe***(st)***
1 for ***i*** = 0 to len***(st)***
2 If ***st[i]*** = ""
3 If ***st[i*** + 1] = 's'
4 Exchange the values of ***st[i*** + 2] and ***st[i*** - 1]
5 ***st[i]*** = "of"
6 Delete ***st[i*** + 1]
7 If ***st[i*** - 2] in English.determiner
8 ***temp*** = ***st[i*** - 2]
9 ***st[i*** - 2] = ***st[i*** - 1]
10 ***st[i*** - 1] = ***st[i]***
11 ***st[i]*** = ***temp***
12 return ***st***

3.5 Inference of statements & Results

This software is based on mining of data from the posts by the potential victims of a flood. Fics does not make the target community fill a strict form unlike other software systems as dicussed in the Literature review section.

Fics allows people to post their status during a flood event as they like it and takes the responsibility to decode their statements in desired mathematical format.

The inference in the string array ***st*** is concluded in two ways, checking for a value given in the string as water height, and checking for the reference objects given in the string.

3.1.1 Checking for Values

Search for units. If a unit in height occurs, store the values as water height. This is done by the function CheckHeightUnits(), described as follows.

CheckHeightUnits(***st***)
1 For each ***word*** in ***st***
2 If ***word*** is a floating point number
3 If ***word*** + 1 is a unit of height (displacement)
4 ***waterHeight*** = Convert the value in SI unit as per the unit referred
5 return ***waterHeight***

3.1.2 Checking for referred objects

It is also possible that the user has no idea of the exact height of water. So he may refer the objects around him to declare current status. This is done by the algorithm described in the function CheckReferredMeasurements(), described as follows.

CheckReferredMeasurements(***st***)
1 For each ***object*** in database of referred objects
2 If ***object.***split() is a subset of ***st***

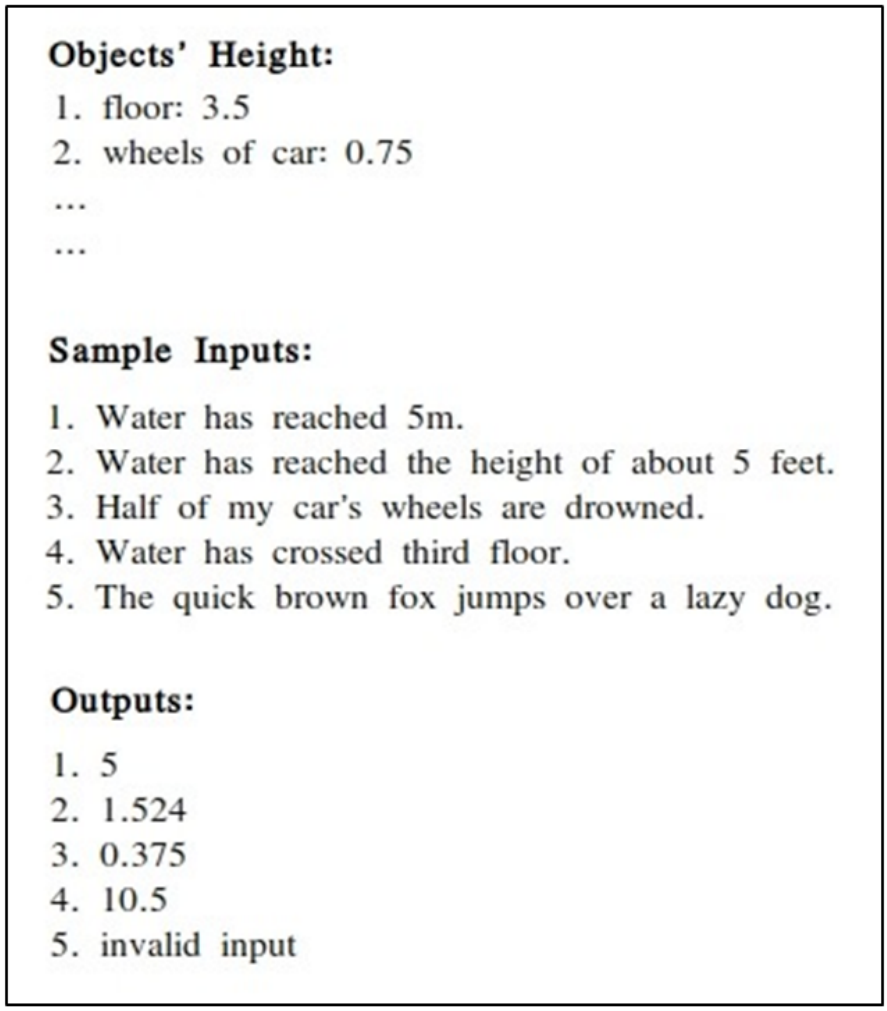

Figure 2. The database of objects' height; the output water heights according to sample inputs and the objects' heights.

3 ***height*** = Fetch the height of object from the database
4 If the object is followed by any degree-word
5 ***degree*** = mathematical form of degree-word
6 ***waterHeight*** = height * degree
7 return ***waterHeight***

Here degree-words which are referred are expect to be following the referred object. Such as "half of car", "second floor" and so on. So the degree-word must be converted into its mathematical form degree and multiplied with water height height to give the then height of water waterHeight.

4.0 Results

We have tested this software system with various synthetic inputs. It has given us apt outputs; a few of the results are shown in Figure 2. Fics is expected to be of great assistance for real-time modelling of flood, and for supply of support system to people in real time at the exact geographical location required. A few sample inputs and related outputs are given in figure 2. The outputs are water heights in their SI unit form. For objects that are referred in the input, Fics refers the database for their related heights. Fics also detects if some fraction (such as "half") or a degree-word (such as "third") is referred; then, it computes the final height which is being referred in the post.

It also translates helping verbs (such as "of"), apostrophes (such as in "car's"), and other helping verbs and words into corresponding mathematical notations.

Fics checks for some keywords (such as "water", "flood", "drown") before starting the computation.

The absence of these keywords will mean that the input is invalid. A similar concept is presented in [14].

5.0 Improvisation of Fics

If we restrict ourselves to only those geotagged posts coming from inside the area of a (probable) flood event, we may expect direct statements which contain the required information in the shortest possible form. Therefore mining of data from those statements will not be of a great issue with respect to artificial intelligence techniques and time involved to process the data.

We may extract the flood status at a particular location easily. Although, we might want to do some changes to refine the results that we receive from raw data.retrieve it.

5.1 The Database

The database as discussed in Fics may be extended to cover all the possible referred objects, verbs, symbols, measurement units and degree-words to cover other possibilities using the same algorithm. This extension of database may depend on the locality of a flood event.

5.2 Linking to a real time system

Fics needs to be linked to a real time system which fetches public posts by a community. Such a system will be able to immediately fetch water heights from the posts and finally model a flood event.

5.3 Non formal results

This software may result in non-uniform non-formal results with respect to water height at geographical locations. This is possible because the values which are obtained come from a non-scientific community. So they may not be exact.

Hydraulic Coherence algorithm discussed in [10] may be applied to these values which will alter the water heights according to the terrain of that geographical locality. Hence we may formalize the values.

6.0 Availability of sample code

Url of repository: https://github.com/ATGupta/FICSv1

Programming language: Python 3

Developers: Arya Tanmay Gupta (Author)

Means for accessing the code:

The line 14 of file **main.py** introduces a variable st. The string stored in this variable is processed and water height is extracted.

This software can be attached to a larger system which fetches posts from social networking sites in real time. The file **main.py** can be called from such a system to generate water height at various geo-locations and map a flood in real time.

The variable ***height*** defined at line 203 of the file process.py contains the height of water that has been extracted from the string which was passed through st introduced at line 14 of main.py.

Conclusion

If we use the phenomenal efficiency of computer intelligence in disaster management and forecasting, we can get results very accurate and fast. Integrating with a modelling software, the software that we made ensures real time modelling of a flood, which can be quite fast. The concept presented in [14] can be used with Fics, before its software structure, to first of all detect if a disaster such as flood has happened. Fics is based on mining of data from the posts on social networking websites and converting those into desired mathematical format. Hydraulic coherence algorithm [10] may be required to apply on the values received from Fics. These values (water heights) can then be combined with the geolocation tagged with the posts to model a flood event. Fics can be integrated with various APIs and modelling platforms to get the posts and model the flood. A machine intelligence based platform like this can be employed to infer data related to other disasters too.

References

[1] Munich RE (2015), NatCatSERVICE Loss events worldwide 1980 – 2014", UNISDR and PreventionWeb.

[2] Benfield, A., Global Catastrophe Recap November (2015), Aon Benfield Analytics | Impact Forecasting in November, 2015. Available on http://thoughtleadership.aonbenfield.com/Documents/20151208-if-november-global-recap.pdf.

[3] McDougall, K. and Temple-Watts, P. (2012), THE USE OF LIDAR AND VOLUNTEERED GEOGRAPHIC INFORMATION TO MAPFLOOD EXTENTS AND INUNDATION, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume I-4, XXII ISPRS Congress in 25 August – 01 September 2012.

[4] Fienen, M. N. and Lowry, C. S. (2012), Social.Water—A crowdsourcing tool for environmental data acquisition, Computers & Geosciences 49 164–169, Elsevier.

[5] Padawangi, R., Turpin, E., Prescott, M. F., Lee, I., and Shepherd, A. (2016), Mapping an Alternative Community

River: The Case of the Ciliwung, Sustainable Cities and Society, Sustainable Cities and Society 20 147-157.
[6] Kutija, Vedrana; Bertsch, Robert; Glenis, Vassilis; Alderson, David; Parkin, Geoff; Walsh, Claire; Robinson, John; and Kilsby, Chris (2014), Model Validation Using Crowd-Sourced Data From A Large Pluvial Flood, International Conference on Hydroinformatics. Paper 415.
[7] https://www.unisdr.org/we/inform/disaster-statistics
[8] J. N. C. Victorino and M. R. J. E. Estuar (2014), Profiling Flood Risk through Crowdsourced Flood Level Reports, 978-1-4799-6541-0/14/, IEEE.
[9] P. Piyawongwisal, S. Handa, L. yu, and A. Samuel (2011), Going Beyond Citizen Data Collection with Mapster: A Mobile+Cloud Real-Time Citizen Science Experiment, IEEE 7th International Conference on E-Science, e-Science 2011, Workshop Proceedings, Stockholm, Sweden.
[10] R. Hostache, P. Matgen, G. Schumann, C. Puech, L. Hoffmann, and L. Pfister (2009), Water Level Estimation and Reduction of Hydraulic Model Calibration Uncertainties Using Satellite SAR Images of Floods, IEEE Transactions on Geoscience and Remote Sensing, Volume 47, Issue 2.
[11] E. Kamar, S. Hacker, E. Horvitz (2012), Combining Human and Machine Intelligence in Large-scale Crowdsourcing, Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), Conitzer,Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.
[12] A. Marcus, E. Wu, D. R. Karger, S. Madden, R. C. Miller (2011), Crowdsourced Databases: Query Processing with People, 5th Biennial Conference on Innovative Data Systems Research (CIDR '11) January 9-12, 2011, Asilomar, California, USA.
[13] P. Dai, Mausam, D. S. Weld (2011), Artificial Intelligence for Artificial Artificial Intelligence", Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence.
[14] M. Imran, C. Castillo, J. Lucas, P. Meier, S. Vieweg (2014), AIDR Artificial Intelligence for Disaster Response, International World Wide Web Conference Committee (IW3C2), WWW'14 Companion, April 7–11, 2014, Seoul, Korea, ACM 978-1-4503-2745-9/14/04.
[15] Wan, Z., Hong, Y., Khan, S., Gourley, J., Flamig, Z., Kirschbaum, D., and Tang, G (2014), A cloud-based global flood disaster community cyber-infrastructure: Development and demonstration, Environmental Modelling & Software 58 86-94.
[16] Qiu, L., Zu, Z., Zhu, Q., and Fan, Y (2017), An integrated flood management system based on linking environmental models and disaster-related data, Environmental Modelling & Software 91 111-126.
[17] www.mturk.com/mturk/welcome
[18] www.galaxyzoo.org/
[19] www.sdss.org/
[20] https://github.com/ATGupta/FICSv1