Software Interestingness Trigger by Social Network Automation

Iaakov Exman, Avihu Harush and Yakir Winograd
Software Engineering Department, The Jerusalem College of Engineering – JCE – Azrieli, Jerusalem, Israel

Keywords: Software, Interestingness, Trigger, Social Networking, Automation, Randomization, Keywords, Application Ontology, Knowledge Discovery, Message Frequency, Analytics, Marketing.

Abstract: Social Networking software is perceived as a fast way to engage people with potential interest in a certain activity. However, interestingness has been defined as a product of a domain *relevance* – arbitrarily determined by a person's tastes – and a *surprise* – determined by how unexpected is an activity relative to average activities in the domain. Therefore, one must manage two unknowns to trigger the desired interestingness: the person tags and the activity description. The person tags are obtained from an application ontology characterizing the chosen domain. The activity description tries to generate surprises by sharpening what differentiates it from conventional activities. This paper describes a software tool based upon a social networking infra-structure and illustrates its quasi-automated usage with inputs of the relevant tags and the surprising activity for a specific domain, viz. marketing of an Event within a conference. Preliminary results are analysed and discussed at length.

1 INTRODUCTION

A Social Network is thought to be a rapid medium to provide information to all its participants, and selectively engage only those with a potential interest in a certain activity. Thus, it can select sub-sets of participants, say for marketing of the referred activity. In order to effectively use a social network for these purposes, we make two assumptions.

A first assumption is that one should have a model of interestingness. Indeed, such a model has been proposed and tested in a previously published work. This will be succinctly described in sub-section 1.1.

A second assumption is that one must automate the social network usage to the maximal possible extent by means of an infra-structure embodied into a suitable software tool. One should leave only a few domain dependent unknowns to be entered as input to the software tool. These include the characterization of the human participants with a potential interest in the desired domain of activity and a description of the designated activity in which we are focusing.

The overall purpose of this work is to test the validity of these assumptions, by means of building a social network software tool, running it on case studies and demonstrating the plausibility of the results.

1.1 An Interestingness Model

In previous work (Exman, 2009) we have defined *interestingness* as a product – or more generally a composition – of two functions: relevance and surprise, the latter being then referred to as unexpectedness.

Relevance is a measure of the fitness of a specific item to the characterization of a domain. For instance, "football" is an entertaining physical activity fitting the "sport" domain. On the other hand, "chess-playing" while an entertaining activity, is **not** considered a "sport", essentially because it does not involve physical efforts.

Surprise is a measure of the distance of a specific item from the average item in a domain. For instance, many "sport" activities involve usage of a "ball", say football, basketball, tennis, ping-pong, etc. But, "Badminton" which is certainly a sport, even with some similarities to tennis, has the unexpected property that it uses a "shuttlecock" instead of a ball. Probably the reader is unaware that a shuttlecock has a crown of feathers!

Formally, the definition of interestingness is expressed in the following generic equation:

$$Interestingness = Relevance * Surprise \quad (1)$$

Exman I., Harush A. and Winograd Y.
Software Interestingness Trigger by Social Network Automation.
DOI: 10.5220/0006591803260333
In *Proceedings of the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (KDIR 2017)*, pages 326-333
ISBN: 978-989-758-271-4

For a better understanding of the interestingness idea, we mention particular implementations of the functions used to actually calculate interestingness. These functions have two sets of keywords as inputs: one set ***C*** characterizes the current activity, the other set ***D*** characterizes the Domain of interest, say by its average activity.

Relevance can be implemented as a *match* function, in which one compares the sets ***C*** and ***D***. The simplest match function outputs a Boolean variable: its value is 1 if at least one keyword is found in both ***C*** and ***D***, otherwise it is zero valued. More sophisticated match functions are possible.

Surprise can be implemented as a *mismatch* function, say by the symmetric difference Δ of the referred sets, calculated as the union $\bigcup$ of the relative complements of these sets:

$$\begin{aligned} Mismatch &= C \Delta D \\ &= (C - D) \bigcup (D - C) \end{aligned} \quad (2)$$

Thus, this particular interestingness is formulated as:

$$Interestingness = \frac{Match * Mismatch}{NormF} \quad (3)$$

The Normalization Factor *NormF* compensates for eventual differences of total number of keywords in the union set $(C \bigcup D)$.

The definition (1) is suitably adapted to the needs of the specific application of this paper, as seen in section 2.

1.2 Social Network Automation

In the original application of the interestingness definition (Exman, 2009) the arbitrary domain ***D*** was fixed for a given person, whose tastes were known, say to be interested in basketball. Therefore there was a single unknown, represented by the variable item ***C***. The outcome of the calculation was a best fitting of ***C*** to the given ***D***.

In the current application of marketing by means of a social network there are two simultaneous unknowns, turning it into a particularly challenging problem of knowledge discovery. One unknown is the person which one wishes to interest in a certain activity. We don't know the different tastes of the variety of participants in the social network. Thus, the arbitrary domain ***D*** for each person may be slightly different.

If our application is marketing of e.g. a Conference event, an industry oriented person may be attracted by technological state of the art lectures to keep him up-to-date, while an academic oriented person may be attracted to the same Conference as an opportunity to publish a paper. Furthermore, an academic oriented person may be in his or her early career stages looking for a doctoral symposium or be a known professor in later career stages, with different interests.

The second unknown is the characterization of the specific activity ***C***. In a same conference one may have research lectures, doctoral symposium, industrial keynotes, posters, and commercial exhibits.

Having simultaneously two kinds of unknowns implies that a software tool to perform marketing in a social network environment should have in its design an automation mechanism. An important capability of this mechanism is to cause random variations among diverse types of potential participants and diverse characterizations of the offered activities. The purpose of these variations is to maximize the interestingness value, i.e. to maximize the product of the relevance and surprise functions.

1.3 Paper Organization

In the remaining of the paper we refer to related work (section 2) introduce the software architecture of Netomation, a tool for social network automation (section 3), describe a Conference Marketing case study (section 4), provide results of experiments with the tool for the referred case study (section 5), and conclude with a discussion (section 6).

2 RELATED WORK

The extensive related literature covers various aspects of this work. We first mention approaches to interestingness, then refer to social network software automation.

Referring to interestingness concepts, a good review to start reading on this topic is (McGarry, 2005) which refers to measures for knowledge discovery. Tuzhilin in (Tuzhilin, 2002), in the invaluable Handbook (Klosgen and Zytkow, 2002), describes various approaches to interestingness, whether pragmatic or foundational.

Specifically referring to the unexpectedness component of interestingness, one finds Padmanabhan and Tuzhilin in (Padmanabhan, 1999) and Piatetsky-Shapiro and Matheus in their work on the Interestingness of Deviations (Piatetsky-Shapiro, 1994).

Exman explicitly formulated Interestingness as a product of a relevance and a surprise (Exman, 2009). Together with co-workers they applied it to a few different applications, such as discovery in the Web of new chemical structures of interest to pharmaceutics (Exman and Pinto, 2010), using tools as described in (Exman, Amar and Shaltiel, 2012).

Regarding social network software automation, one should be first aware of the ubiquity of bots (short for software robots) in social networks. This is seen e.g. in (Boshmaf et al., 2011). See also several chapters of the book on Twitter and Society (Weller et al., 2014) in particular the chapter on Twitter Accounts by (Mowbray, 2014) describing Twitter bots for marketing.

Shop-bots are internet agents that give buying advice to online shoppers about products and their best price. In a sense these are agents somewhat similar to those which offer marketing suggestions. An analysis by comparison of three models of shop-bot use on the web is provided by (Gentry and Calantone, 2002).

Another issue of importance is the ability to distinguish bots from humans and make classifications of bots. Chu and co-workers (Chu et al., 2010) explicitly ask the question of who is behind the messaging. The same research group (Gianvechio et al. 2008) tries to classify participants in chats.

Recent works dealing with bots recognition include (Gilani et al., 2017) in which an in-depth characterization of Bots and Humans, finds reliable classification despite, some surprising behavior similarities between bots and segments of the human population. Similar results are found in (Ferrara, 2016).

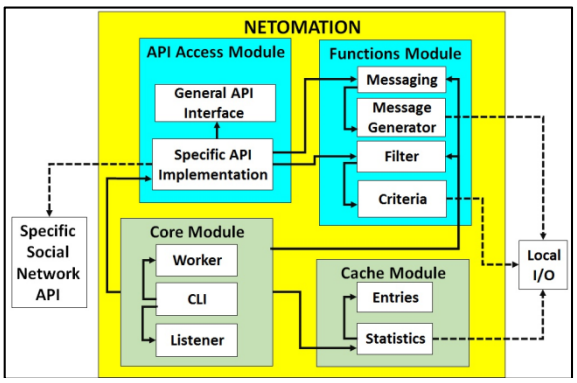

Figure 1: Netomation Schematic Software Architecture – Important upper modules (in blue) are: a- *API*: enables access to social network APIs; b- *Functions*: message and filter related. Supporting lower modules (in gray) are: c- *Core*: basic interaction with Social Networks. d- *Cache*: I/O fast storage. Arrows point to called functions. Dashed arrows link to external units.

Exman and co-workers made an exploration of the survivability of a bot within a social network of humans (Exman, Alfassi, and Cohen, 2012), which is comparable to a sort of (anti)-Turing test.

In this work we take a pragmatic benign and utilitarian approach to social network automation.

3 NETOMATION: A SOCIAL NETWORK AUTOMATION TOOL

Here we shortly describe the software architecture of "Netomation", a social network software tool, and mention its main modules.

3.1 Netomation Software Architecture Separation Principle

The main principle behind the Netomation software architecture is to separate generic mechanisms from the interactions with any specific social network.

The goal is to make any social network replaceable by any other one with minimal changes of Netomation components. This separation principle is detailed as follows:

1. *Generic API access* – this is a set of generic functions common to all social networks, extracted from their respective APIs and wrapped to access any social network, in contrast with API functions specific to a given social network;
2. *Generic Netomation functions* – these are application related message and filter functions, specialized for Netomation, and usable with any social network.

3.2 Netomation Modules Design and Implementation

The Netomation architecture is schematically displayed in Fig. 1. Its main modules are:

1. *API* – its functions enable access to social network APIs, carefully separating generic from specific functions, according to the previous software architecture principle;
2. *Functions* – contains specialized functions to generate messages, frequency randomization and filters;
3. *Core* – it contains basic active (worker) and passive (listener) ways of interaction with

the underlying social network through its API;

4. *Cache* – is the place where input data are supplied and minimal results are stored for run-time I/O and fast saving for posterior data analysis.

Netomation has been implemented in the Java language. It runs in a client-server environment.

Special attention has been given to the efficiency of the cache. For instance, instead of saving the whole information of the messaging target, in real time just the target's identification number is stored. This number can be later used to retrieve additional data for posterior analysis of the results.

3.3 Automation Mechanism

The automation mechanism is defined by a set of conditions including:

a- ***Basic Frequency*** – messages are sent with a basic frequency Bf; for instance, Bf may have a cycle of ***Bc*** hours, typically of the order of 24 hours;

b- ***Randomization*** – on top of the basic frequency there is a randomization ***R*** with a smaller order of magnitude, i.e. the actual cycle ***Ac*** is equal to ***Bc*** + ***R***;

c- ***Message Variation*** – any message is sent only once to each target.

4 CASE STUDY: CONFERENCE EVENT MARKETING

As a case study, to test our assumptions formulated in the introduction of this paper, and to measure whether the tool brings practical benefit, we applied the tool to the marketing of a Conference Event, within a commercial social network.

Following the ideas of the interestingness model of sub-section 1.1 we first characterize the people who are the marketing target and then the Conference Event activities being marketed.

4.1 Characterization of the Marketing Target: Application Ontologies

As the Conference Event is science-oriented, the potential audience is composed from students, academics, workers in the hi-tech industries and in research laboratories. The potential audience is not uniform and variability should be taken into account.

Thus, the application ontology characterizing the audience, needed to calculate the *relevance* in equation (1), is obtained from selected terms from more than one domain ontology. This is illustrated by keywords from say three domains:

a- software related engineering disciplines;
b- academic institutions;
c- hi-tech industries;

Two of these partial application ontologies are displayed in Fig. 2 for software related keywords and in Fig. 3 for academic institution keywords.

#	1st level terms	2nd level terms	3rd level terms	4th level terms
1	**Discipline**			
2		Computer Science		
3		Software Engineering	→ Software	
4		Systems Engineering	→ Systems	
5		Knowledge Engineering	→ Knowledge	→ Ontology
6	**Activity**			
7		Requirements		
8		Design		
9		Development	→ Agile	
10		Implementation	→Programming	
12		Running		

Figure 2: Software Related Keywords – This is a partial application ontology displayed as a table of terms referring to software related engineering disciplines and software development activities. Arrows have the usual meanings as in ontologies (either subtype or composition), e.g. Agile is a sub-type of Development.

#	1st level terms	2nd level terms	3rd level terms
1	**Academic Institution**		
2		University	→ School
3		Institute	→ Faculty
4		College	
5		Research	
6	**Roles/ Degrees**		
7		Professor	→ Prof
8		Doctor	→ Dr
9		Lecturer	
10		Scholar	
12		Scientist	
13		Programmer	→ Programming

Figure 3: Academic Institution Keywords – Another partial application ontology referring to academic institutions and roles/degrees of people working/studying in these institutions. Arrows have the usual meanings as in ontologies (either subtype or composition), e.g. an Institute may be composed of Faculties.

4.2 Characterization of the Conference Event Activities: Typical Messages

A pre-defined set of the order of magnitude of 10 messages was formulated as input for our software tool to use as part of its regular operation. The relevant URL was appended as an additional information item.

The characterization of the Conference Event activities is reflected in the surprising messages sent to the potential audience. Two examples are as follows:

Example Message 1:

"Software is Knowledge and Knowledge is Power: Event-Name, Submit paper by September 4"

Example Message 2:

"Event Name: Concepts are the 'atoms' of Software Knowledge, Submit paper by Sept 4"

Each of the messages have three distinct parts:

a- ***Event name***: The Conference Event;
b- ***Action to be taken***: Submit paper by date;
c- ***Surprise Sentence***: 'Concepts are atoms…'

The surprise sentence usually links terms in the application ontology – e.g. Knowledge, concepts – with a somewhat unrelated term and therefore contributing to the *surprise* in equation (1) – e.g. Power, atoms.

Moreover, each sentence was modified to create a few variants such that the social network server will not flag program messages as potential spam.

5 PRELIMINARY EXPERIMENTS

In order to test the assumptions behind our tool and its effectivity by means of the outcomes, we have performed a series of experiments. The execution of the experiments continues in this ongoing project.

Here we display results of preliminary experiments performed during a series of consecutive days, just before the paper writing. The experiments collected data about targets, which our tool followed, our followers and messages received from some of the targets. The followers and messages received are shown both as time-dependent results and as geographical data.

5.1 Time-Dependent Results

The time-dependent curve showing the number of followers is displayed in Fig. 4 for 24 consecutive days, in the dates marked in the horizontal axis of the figure.

The number of messages received increased during the initial period of four days with the same general trend as the followers, but slightly faster. After this initial period, the rate of messages received decreased.

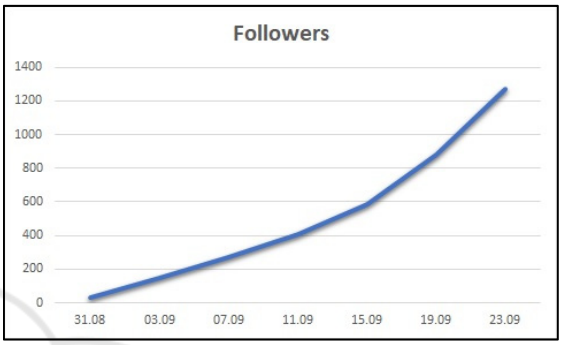

Figure 4: TIME DEPENDENT RESULTS – Cumulative number of followers during 24 days from August 31, until September 23, in 2017. The vertical axis is the number of followers. The horizontal axis shows dates in a numerical format of "***day.month***". The graph is an estimated fitting.

5.2 Geographical Data

The geographical distribution of followers is seen in Fig. 5. The total number of followers from all countries in this 24 days period was 1341. The majority of these results come from the USA and Great Britain. Other countries include:

- Asia: Bangladesh, China, Hong-Kong, India, Indonesia, Iran, Israel, Japan, Korea, Malaysia, Pakistan, Philippines, Saudi Arabia, Singapore, Turkey, UAE;
- Europe: Belgium, Denmark, Finland, France, Germany, Greece, Italy, Netherland, Poland, Portugal, Russia, Slovenia, Spain, Sweden, Ukraine;
- Other African: Algeria, Congo, Egypt, Ethiopia, Ghana, Kenya, Nigeria, South Africa, Uganda, Zambia, Zimbabwe;
- Other American: Argentina, Brazil, Canada, Chile, Colombia, Mexico, Panama, Paraguay, Peru, Uruguay, Venezuela;
- Other Oceania: Australia and New Zealand;

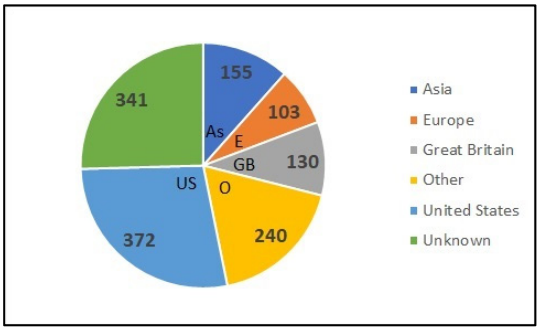

Figure 5: Geographical Distribution of Followers – The majority of followers are from the US and GB (Great Britain). The "O = Other" rubric includes countries with small numbers of followers. "Unknown" means unavailable country information.

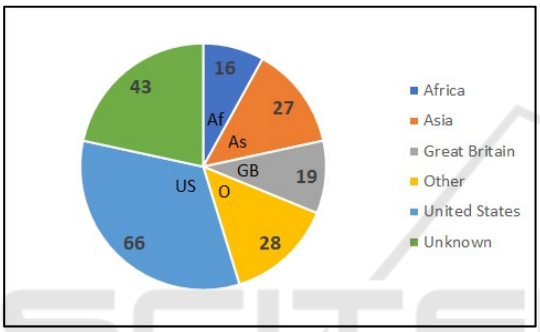

Figure 6: Geographical Distribution of Sources of Messages Received – It shows that the majority of messages were received from the USA, GB (Great Britain) and India (included in Asia). The same conventions are used as in Fig. 5.

The geographical distribution of sources of messages received is seen in Fig. 6. The total number of message sources in this 24 days period is 199. The overwhelming majority in these results comes from a few countries, similarly to the followers in Fig. 5. Other countries include:

- Asia: Bangladesh, China, Hong-Kong, India, Indonesia, Japan, Mongolia, Pakistan, Saudi Arabia, Sri Lanka, Thailand
- Europe: Belgium, France, Germany, Italy, Macedonia, Slovenia, Spain, Sweden, Ukraine
- Other African: Egypt, Ghana, Kenya, Nigeria, Tanzania, Zimbabwe
- Other American: Argentina, Brazil, Canada, Guyana, Mexico, Paraguay, Venezuela,
- Other Oceania: Australia

Please note that there is a significant overlap between the geographical distributions of the list of countries of our followers (corresponding to Fig. 5) and the list of message sources (corresponding to Fig. 6), but these lists are neither identical nor one is a subset of the other.

6 DISCUSSION

This discussion examines the obtained preliminary results. Then it deals with foundational issues, pragmatic considerations and future work. It is concluded with a short statement of the main contribution and our wider vision on this work.

6.1 Examination of Preliminary Results

One could summarize the obtained preliminary results with respect to its three aspects.

Concerning time-dependence:

a. ***General Trend*** – the general trend of increasing numbers of followers and of numbers of received messages with time is reasonable for the initial experimentation period; much longer periods should be tested in order to analyse in depth the Netomation behavior.

b. ***Slope of Increase with Time*** – graphs of followers (shown in Fig. 4) and received messages increase faster than linearly; the specific values of graph slopes needs more experimentation in order to be understood and to check whether one can exert a desirable control over it;

Concerning geographic distribution of countries:

a. ***Bias*** – there is a clear overall bias towards English-speaking countries; this deserves further investigation, since its source is not absolutely clear; one could perhaps link the size of some of the countries (USA and India), but this does not explain the case of Great Britain; once one formulates an hypothesis about the source of this bias, it will be tested, to allow better control of the automatic application steering;

b. ***Partial Information*** – the number of "Unknown" locations both of followers in Fig. 5 and sources of received messages in

Fig. 6 is non-negligible (more than 20%); it is possible to reduce the number of unknowns and improve the information obtained by means of a more extensive investigation, besides the direct data from this variable provided by the social network.

Concerning randomization:

a. ***Event Contingencies*** – it is important to stress the reasons behind and the contingencies of the randomized nature of communicating with a social network server, as many parameters and environment variables are unknown. Netomation had to wait arbitrary numbers of seconds before every action it performed, since from time to time the social network server did mark Netomation operations as "spam". It forced the program to either wait for the warning to fade or to skip to other actions.

b. ***Human Beings vs. Bots*** – the undisclosed nature of human beings, which means designing an efficient filter to identify potential target audience – is a given follower a human being or a bot? – can be challenging to say the least.

6.2 Foundational Issues

Foundational issues refer first and foremost to the plausibility of the assumptions made for this work (stated in the Introduction section of this paper), viz.:

a. ***Interestingness Model*** – is the model adequate for the kind of application described in this paper?

b. ***Tool Automation Nature*** – is the separation boundary between the generic tool automation from specific data (referring to a particular social network and to a particular application, such as marketing) adequate with respect to software architecture?

Regarding both of these issues we are in a too early stage of experimentation within this project, in order to provide clear-cut and definitive answers.

In order to answer the first question one needs to explicitly calculate the values of the relevance and surprise functions from the tool inputs and check their correlation with output results.

In order to answer the second question one needs to use the tool with different social networks and provide success criteria to be checked.

6.3 Pragmatic Considerations

Pragmatic considerations involve a long series of issues, referring among other things to:

a. ***Social Network API*** – the comprehensive and generous support of the social network API (Application Programming Interface) should be of interest for both users and social networks themselves, as it opens wide horizons to newer kinds of applications; we mention here e.g. Rest API and language specific support;
b. ***Software Efficiency*** – is the time and memory resources utilization efficient in practice? In particular we mean the storage of only identification (id) numbers and posterior off-line retrieval of further data based on these id numbers to perform analysis.
c. ***Application Success*** – is the approach using social networks in the proposed general lines important, necessary but not sufficient or powerful enough for the chosen kind of application? Is it useful for other related applications?
d. ***Biases and Filters*** – how easy it is to automatically steer the software tool, by means of filters, in order to correct eventual biases, and to focus, on say specific desired countries, within marketing applications?

6.4 Future Work

There is still much work to be done in this ongoing project. Some of the most important open issues are:

a. ***Explicit Interestingness Calculation*** – for given inputs and check their correlation with application outputs; comparison should be made between an application with and without usage of "surprises";
b. ***Extensive Data*** – the preliminary data in this paper refer to a relatively short time period; we need to run the tool for longer periods, while experimenting with different inputs and filters;
c. ***Diverse social Networks*** – in order to test the actual generality and robustness of Netomation, one needs to use it with a few

different social networks and compare their behaviours and results;

d. ***Receiver Automation*** – in the current work we emphasize what we learned from "sender" automation; it seems that there is much opportunity of work to be done in the complementary "receiver" automation.

6.5 Main Contribution

The main immediate contribution of this work is the separation principle within the software architecture of social network automation for practical application purposes. In the long term, it opens horizons for newer kinds of applications and a wider vision for social networks.

REFERENCES

Boshmaf, Y., Muslukhov, I. Beznosov, K. and Ripeanu, M., 2011. "The Socialbot Network: When Bots Socialize for Fame and Money", in Proc. ACSAC'11 27th Annual Computer Security Applications Conf. pp. 93-102. DOI: 10.1145/2076732.2076746

Chu, Z., Gianvecchio, S., Wang, H. and Jajodia, S., 2010 "Who is Tweeting on Twitter: Human, Bot or Cyborg?", in Proc. ACSAC'10, 26th Annual Computer Security Applications Conf. pp. 21-30. DOI: 10.1145/1920261.1920265

Exman, I.. 2009. "Interestingness – A Unifying Paradigm – Bipolar Function Composition", in Proc. KDIR Int. Conf. on Knowledge Discovery and Information Retrieval, pp. 196-201, DOI: https://doi.org/10.5220/0002308401960201

Exman, I. and Pinto, M., 2010. "Lead Discovery in the Web", in Proc. KDIR Int. Conf. on Knowledge Discovery and Information Retrieval, pp. 471-474, DOI: https://doi.org/10.5220/0003098304710474

Exman, I., Alfassi, N. and Cohen, S., 2012. "Semantics of Social Network Frequencies for Turing Test Immunity", in Proc. SKY'2012 Int. Workshop on Software Knowledge, pp. 79-84. DOI: 10.5220/0004181600790084

Exman, I., Amar, G. and Shaltiel, R., 2012. "The Interestingness Tool for Search in the Web", in Proc. SKY'2012 Int. Workshop on Software Knowledge, pp. 54-63. DOI: 10.5220/0004178900540063

Ferrara, E., Varol, O., Davis, C., Menczer, F. and Flammini, A., 2016. "The rise of social bots", Comm. ACM, Vol. 59, pp. 96-104. DOI: 10.1145/2818717

Gentry, L. and Calantone, R., 2002. "A comparison of three models to explain shop-bot use on the web", Psychology and Marketing, Vol. 19, pp. 945-956. DOI: https://doi.org/10.1002/mar.10045

Gianvecchio, S., Xie, M., Wu, Z. and Wang, H., 2008. "Measurement and classification of humans and bots in internet chat", In Proc. 17th USENIX Security Symposium, San Jose, CA.

Gilani, Z., Farahbakhsh, R., Tyson, G., Wang, L. and Crowcroft, J., 2017. "An in-depth characterization of Bots and Humans on Twitter", https://arxiv.org/pdf/1704.01508.pdf

Klosgen, W. and Zytkow, J.M. (eds.), 2002. *Handbook of Data Mining and Knowledge Discovery*, Oxford University Press, Oxford, UK.

McGarry, K., 2005. "A survey of interestingness measures for knowledge discovery", Knowledge Engineering Review J., 20 (1), 39-61. DOI: https://doi.org/10.1017/S0269888905000408

Mowbray, M., 2014. "Automated Twitter Accounts", Chapter 14 in ref. (Weller et al., 2014), pp. 183-194.

Padmanabhan, B. and Tuzhilin, A., 1999. "Unexpectedness as a measure of interestingness in knowledge discovery", *Decision Support Sys*., Vol. 27, (3).

Piatetsky-Shapiro, G. and Matheus, C.J., 1994. "The Interestingness of Deviations", KDD-94, AAAI-94 Knowledge Discovery in Databases Workshop.

Tuzhilin, A., 2002. "Usefulness, Novelty, and Integration of Interestingness Measures", chapter 19.2.2 in reference (Klosgen and Zytkow, 2002), pp. 496-508.

Weller, K., Bruns, A., Burgess, J., Mahrt, M. and Puschmann, C. (eds.), 2014. *Twitter and Society*, Peter Lang Publishing, New York, NY, USA.