Automatic Audience Focusing by Event Interestingness

Iaakov Exman, Yakir Winograd and Avihu Harush
Software Engineering Department
The Jerusalem College of Engineering – JCE - Azrieli
Jerusalem, Israel
iaakov@jce.ac.il, yakirwin@gmail.com, tchvu3@gmail.com

***Abstract*— Large social Networks have marketing potential to spread information about interesting events to suitable audiences. However, huge network sizes and varieties of information available are obstacles to reach the desired goal. This paper investigates the hypothesis of computable Interestingness as a criterion to focus on suitable audiences for any given event. Interestingness is calculated by combining two functions: Relevance and Surprise. A generic software tool has been developed as an experimental testbed to interact with any social network. Its inputs are the event characterization and audience candidates for the given event. Two results validate this work's hypothesis: first, audience candidates who actually visited the event site, have on the average a bigger computed Interestingness than the rest of the population; second and most important, computed Interestingness better differentiates event site visitors, actually interested in the Event, from non-visitors, while Relevance alone, does not distinguish so-well between visitors and non-visitors.**

Keywords: Interestingness; Automatic focusing; Event; Relevance; Surprise; Match; Mismatch; Randomization; Social Network; Software Architecture; Knowledge Discovery; Analytics; Marketing.

I. INTRODUCTION

Social Networks, by labeling their members with common interests, have the potential of efficiently marketing specific events. On the other hand, their huge sizes and proliferation of information types are impediments to reach the desired goals.

Our working hypothesis is that a computable Interestingness criterion focuses on suitable audience candidates for a chosen event, overcoming the sizes obstacle. Having a computable Interestingness, one automates its usage with a software tool. This tool is an experimental testbed for the working hypothesis.

The goal of this paper is to validate the working hypothesis by comparison of the calculated Interestingness with the behavior of audience candidates. The validation experiment consists of: 1) Send information items to audience candidates; 2) Compute the candidate's interestingness relative to the given event; 3) Compare it with the candidate action of visiting or not the Event Web site.

DOI: 10.18293/SEKE2018-220

II. RELATED WORK

We concisely review the literature related to Interestingness, its applications, and internet agents within social networks.

A. Interestingness Concepts and Applications

Overviews of Interestingness measures for Data Mining and knowledge discovery are given by Geng et al. [6] and McGarry [11]. Interestingness approaches are described by Tuzhilin [13] in the Klosgen and Zytkow Handbook [9].

Criteria to determine interesting rules/patterns generated in data mining are discussed by Lenca et al. [10].

Exman, in 2009, [2] defined Interestingness as a product of relevance and surprise. This definition has been embodied in Web search software tools such as the one described in [4].

B. Social Networks Applications

Social network bots, i.e. software robots, are ubiquitous, as seen in the book on Twitter and Society by Weller et al. [14]. Of particular interest is the Twitter Accounts chapter by Mowbray [12], describing Twitter marketing bots. Gentry et al. [7] analyze shop-bots, advising online shoppers about products and prices.

It is essential to distinguish bots from humans. Chu et al. [1] deal with this issue. Gilani et al. [8] also aim at bot recognition. Ferrara [5] reliably classify bots despite similar behavior to humans. Exman et al. [3] explored bot survivability within a human social net, as a kind of anti-Turing test.

III. INTERESTINGNESS

Here we give generic and specific Interestingness definitions. Then, events and candidates are characterized.

A. Interestingness Definitions

The assumptions behind the Interestingness definition are:

1. ***Domain of interest choice is arbitrary*** – one may express interest in pre-Columbian archeology or in Jazz music; any choice is a matter of personal taste;
2. ***Unusual items attract more attention than average items*** – unusual items should be given more weight than average ones, when computing interestingness.

Interestingness is a two function composition: the *Relevance* of an item to a domain chosen by one's personal taste and the *Surprise* caused by most unusual items, among the relevant ones. We simplify it to be just a commutative multiplication:

$$Interestingness = Relevance * Surprise \quad \textbf{(1)}$$

In this work an item is a candidate for a conference event, in a given domain. *Relevance* measures to what extent the candidate fits the event audience. *Surprise* measures to what extent the candidate for a conference event is outstanding, relatively to the average candidate for this event.

There exist several specific functions to calculate Relevance and Surprise. A well-known formula is ***TfIdf*** used in data mining. ***Tf***, Term Frequency, expresses Relevance, based upon the chosen term frequency in a given document. ***Idf***, Inverse Document Frequency, expresses Surprise, or rarity, i.e. inverse ratio of relevant documents relative to all examined documents.

In this work ***match*** and ***mismatch***, respectively stand for Relevance and Surprise. These functions compare keyword sets for each Candidate ***C*** with the keyword set for Event ***E***. ***Match*** calculates a similarity measure of the input sets, i.e. keywords appearing in the intersection $\bigcap$ of these sets:

$$Match = C \bigcap E \quad \textbf{(2)}$$

The output is the number of intersection elements of ***E*** and ***C***.

Mismatch calculates the sets' dissimilarity, viz. a symmetric difference Δ between ***E*** and ***C***. It is the union $\bigcup$ of the relative complements of these sets:

$$Mismatch = C \Delta E = (C - E) \bigcup (E - C) \quad \textbf{(3)}$$

The final formula is normalized by a factor NormF to assure results independence of set sizes:

$$Interestingness = \frac{Match * Mismatch}{NormF} \quad \textbf{(4)}$$

B. Characterization of Events and Candidates

The keyword sets characterizing an event are obtained from its Call-for-Papers after filtering stopwords, i.e. frequent words such as conjunctions and articles, "and", "the", which are not domain specific. Candidate characterization is similarly obtained, and schematically seen in Fig. 1.

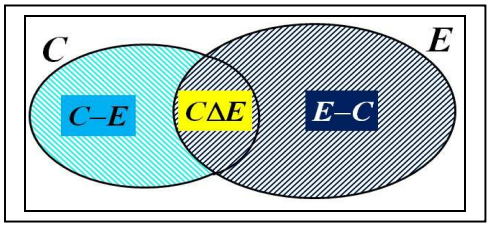

Figure 1. Schematic Match and Mismatch diagram – ***E*** is the Event set (dark blue). ***C*** is the candidate set (light blue). Match is the intersection ***C*Δ*E*** (in yellow). Mismatch is the union between the relative complements ***C*-*E*** and ***E*-*C***.

IV. The AutoFocus System Software Architecture

The AutoFocus software tool aims to automatically test the focus on targets within an event by computing Interestingness.

A. Overall Experimental Client-Server System

The experimental system client-server architecture (in Fig. 2) enables server interaction with the remote user agent through the Web. The server interacts through a Restful API with the AutoFocus tool, each having its own database.

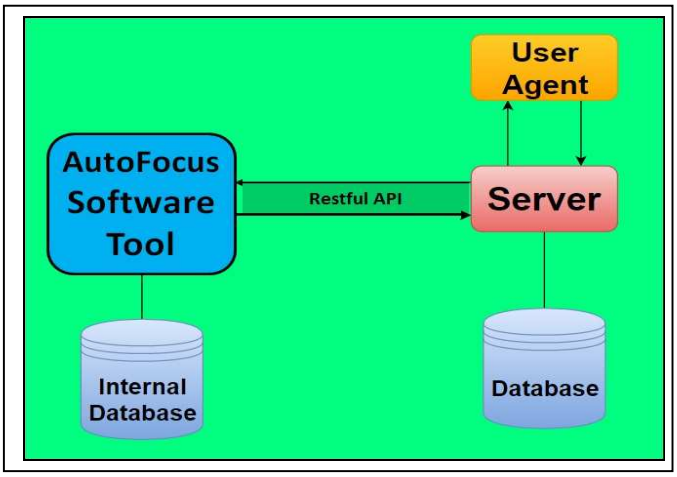

Figure 2. Overall Experimental system client-server Software Architecture – This system has three components: a *User client agent* and its *Server* and the system core, the *AutoFocus tool*. Both the Server and the AutoFocus tool have their own Database, and they communicate through a Restful API.

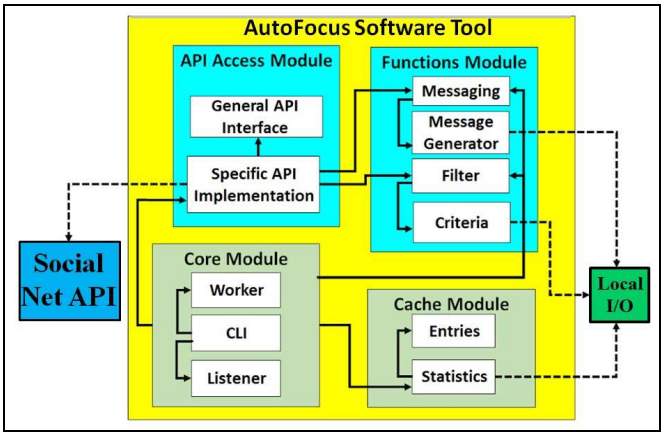

Figure 3. AutoFocus Software tool Architecture – The main sub-system (in yellow) has four modules. Two infrastructure modules: Core and Cache; two essential modules: generic API Access and the Functions module. Outside the main sub-system we emphasize the Social Net API for any specific Social Net.

B. AutoFocus Tool Generic Architecture

The architecture goal is to clearly separate a generic API interface from any specific social net API. Assuming that social networks have much in common, one replaces any specific social net attached to the AutoFocus tool by any other one, with minimal or no modification of the generic API interface. The generic Functions Module, is also usable with any social net.

The AutoFocus architecture, in Fig. 3, is composed of infrastructure (Core and Cache Modules) and essential

functionalities (General API to access any social network and Functions Modules). The AutoFocus tool is implemented in Java. The Interestingness calculator is programmed in Python.

C. Automation Criteria

Messages are automatically sent to social net members by some basic criteria: a) ***ground frequency*** (e.g. once in 24 hours) upon which actual communication is randomized; b) ***random latency*** with a lesser order of magnitude than the ground frequency (e.g. order of minutes); c) ***message variation*** specific message contents are sent only once to each target.

V. Results and Discussion

Starting from a small initial candidate set, the AutoFocus tool scans the net searching for new members related to previous candidates; the new members are added to the candidate list and the process continues recursively. Messages are actively sent and passively received, while calculating Interestingness values.

A. Geographic Distribution Results

The countries distribution for a sample of AutoFocus contacts (received/sent messages) is seen in Fig. 4. These are: a) ***Asia*** – 8 countries, 8 contacts; b) ***Europe*** – 14 countries, 53 contacts; c) ***North America*** – Canada, USA, 34 contacts; d) ***Other*** – from Africa, Oceania and South America.

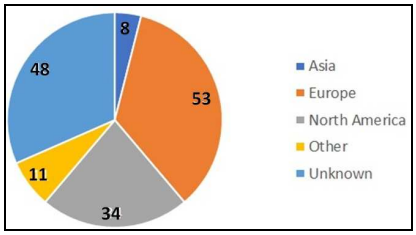

Figure 4. Geographical Distribution of Contacted Candidates – The majority of contacts are from Europe and North America. "Other" means a few countries in other continents. "Unknown" means unavailable country information.

B. Interestingness vs. Event Site Visitors

Fig. 5 shows statistics for a larger sample of 959 candidates, "visitors" (those that visited the Event Site) vs. "non-visitors".

	Visitors	Non-Visitors
Candidate Percentage	9.59%	90.41%
Average *Interestingness*	0.255	0.154
Average *Matched_Only*	0.087	0.053

Figure 5. Candidate Statistics for Visitors vs. Non-Visitors.

C. The Importance of Surprise in Interestingness

Preliminary conclusions from the above results are:

a- A significant number of candidates visited the Event Web site. Their average *Interestingness* is clearly higher than the average of those that did not visit the Event site;

b- The average *Matched_Only* does not distinguish so-well visitors from non-visitor candidates: *Surprise* within *Interestingness* is significant.

Typical search techniques look for similarities with a pattern. The Relevance function does exactly this. This is feasible with a comparison standard. But, when searching something potentially interesting, and not sure about its existence, or without an available standard, using Relevance alone is infeasible.

The importance of ***Interestingness*** for searching – either in the Web or in other large data depositories – and for focusing on candidates for a certain Event resides in the ***Surprise*** function. This work's experiment points out to the feature, that even when a standard ruler is available, Interestingness – including Surprise – affords advantages in order to focus on suitable items.

D. Future Work & Main Contribution

Future work includes: time-axis distribution; measuring larger samples; usage of other Interestingness expressions such as TfIdf; more precise statistical criteria for analysis; experiments with other events.

The main contribution of this paper, besides AutoFocus tool generic development, is to test the independently computed Interestingness as a criterion to focus on candidates with real interest in a certain event, viz. Event Web site *visitor* candidates.

References

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

[2] I. Exman, "Interestingness – A Unifying Paradigm – Bipolar Function Composition", in Proc. KDIR Int. Conf. on Knowledge Discovery and Information Retrieval, pp. 196-201, 2009.

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

[4] I. Exman, G. Amar and R. Shaltiel, R., "The Interestingness Tool for Search in the Web", in Proc. SKY'2012 Int. Workshop on Software Knowledge, pp. 54-63, 2012.

[5] E. Ferrara, O. Varol, C. Davis, F. Menczer and A. Flammini, "The rise of social bots", Comm. ACM, Vol. 59, pp. 96-104, 2016.

[6] L. Geng and H.J. Hamilton, "Interestingness Measures for Data Mining: A Survey", ACM Computing Surveys, Vol. 38, (3), Article 9, 2006.

[7] L. Gentry and R. Calantone, "A comparison of three models to explain shop-bot use on the web", Psychology and Marketing, Vol. 19, pp. 945-956, 2012.

[8] Z. Gilani, R. Farahbakhsh, G. Tyson, L. Wang and J. Crowcroft, "An in-depth characterization of Bots and Humans on Twitter", 2017.

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

[10] P. Lenca, P. Meyer, B. Vaillant and S. Lallich, "On selecting interestingness measures for association rules: user oriented description and multiple criteria decision aid", European J. Operational Res., Vol. 183, pp. 610-626, 2008.

[11] K. McGarry, "A survey of interestingness measures for knowledge discovery", Knowledge Engineering Review J., 20 (1), 39-61, 2005.

[12] M. Mowbray, "Automated Twitter Accounts", Chapter 14 in ref. [21], pp. 183-194, 2014.

[13] A. Tuzhilin, "Usefulness, Novelty, and Integration of Interestingness Measures", chapter 19.2.2 in ref. [14], pp. 496-508, 2002.

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