Semantics of Social Network Frequencies for Turing Test Immunity

Iaakov Exman, Netanel Alfassi and Shani Cohen

Software Engineering Department, The Jerusalem College of Engineering,
POB 3566, Jerusalem, 91035, Israel

Abstract. People using social networks make an implicit assumption that other players in the network are themselves human. But, while in most cases this assumption could be valid, we claim that this is a rather challenging example of a Turing test for artificial intelligence. To verify this claim we have designed and actually built a CyboPlayer – a software package that has the expected behaviour of a human player in the network. The important characteristics of the design include functions to perform in the social network and adjustable parameters to trigger reasonable behaviors. The final Turing test has been to liberate CyboPlayer in the network, like a bird in nature, and observe its interactions with other players. The paper describes the design, implementation and preliminary results of experiments.

1 Introduction

Social networks, like Facebook, LinkedIn or Twitter, are a recent rapidly spreading phenomenon in the Web. They have taken a significant fraction of the leisure or even professional time of people.

Persons, using social networks and proposing friendship to other players in the network, usually make – naively or not – the assumption that other players in the social network are themselves humans. This assumption may be in most cases valid. On the other hand, there are known cases in which there is deliberate manipulation of impersonated players to achieve goals extraneous to the average network user.

In this work, instead of looking at the impersonated players from a security point of view, as a nuisance to be avoided, we have taken the position that they are an interesting and challenging example of the Turing test for artificial intelligence (AI) devised by Turing in 1950 [6].

Thus, the main issues of this paper are the characteristics that make the AI player indistinguishable from human players.

We decided to investigate these characteristics by actual design and implementation of a software package – the CyboPlayer – and subsequent test of its behaviour by liberating it in its "natural environment" viz. the social network.

1.1 Related Work

Here one finds a very concise literature review.

Exman I., Alfassi N. and Cohen S..
Semantics of Social Network Frequencies for Turing Test Immunity.
DOI: 10.5220/0004181600790084
In *Proceedings of the 3rd International Workshop on Software Knowledge* (SKY-2012), pages 79-84
ISBN: 978-989-8565-32-7

The usage of non-human players in social networks is widely known. For instance, Chu and collaborators [3] deal with various types of software players – classified as bots or cyborgs – in Twitter. Gianvecchio et al. [4] refer to measurement and classification of bots in internet chat, from a security point of view. Boshmaf et al. [1] deal with socialbots whose purpose is financial and marketing profits.

Yan [8] contrasts bots and cyborgs' behavior with respect to the security mechanism known as CAPTCHA (Completely Automated Turing Test to Tell Computers and Humans Apart).

Analysis of interaction frequencies and structure of communities in social networks has been extensively done. For instance, Newman [5] analyses community structures in networks, with techniques that are applicable to social networks.

In the remaining of the paper we introduce the functionality of cyborgs for social networks (section 2), describe semantic approaches to interaction frequencies (section 3), overview the Cyboplayer software architecture (section 4), discuss validation techniques (section 5) and conclude with a discussion (section 6).

2 Functionality of Cyborgs for Social Networks

The CyboPlayer of this paper is a kind of cyborg, i.e. a human–assisted bot. CyboPlayer is ultimately intended to be automatic, but allows for human intervention during experiments.

This section describes its functionality. It can be roughly classified as either independent actions or interaction/friendship related.

2.1 Independent Actions' Functionality

A typical partial list of independent actions is:

- *Event generation* – an event has access rights (private, public or group restricted), a name and a given location;
- *Post/status publication* – the post has access rights (all, friends, friends of friends), identifier number and contents;
- *Upload picture* - a picture is published in the personal wall; the picture has an identifier number, link to the picture, tags on people in the picture, upload date;
- *Get-my-friends* – this is a function returning the current list of own friends (names, identifier) in a dynamic object;
- *Get-user-friends* – sends a message to an user, which returns his current list of friends (as in previous function).

Note that Get-my-friends and Get-user-friends are isolated actions and not interactions with friends. These are functions returning lists.

2.2 Interactions/Friendship Related Functionality

Interactions are reactions to initiatives of other players in the network. Friendship related functions involve other players. Here is a typical partial list of such functions:

- *Like* – a positive reaction (without contents) to an initiative of another player, with object identifier and player identifier;
- *Comment* - a reaction with contents;
- *Send Friend Request* – send a request of friendship to another player (has player identifier);
- *Check Friend Request* – checks if received a friend request from another player and confirms it.

3 Semantic Approach to Turing Test Immune Cyborgs

The immunity of a cyborg relative to Turing tests is given first of all by a reasonable series of independent actions and reactions. In other words, the frequencies of actions/reactions are important parameters in this play.

Two central ideas regarding frequencies are:

- *Frequencies are Bounded* – mainly from above and to a lesser extent also from below; a too high frequency of events is suspicious of not being originated by humans;
- *Randomize Specific Event Timings* – frequencies cannot be strictly repetitive.

We claim that a better understanding of reasonable series of actions/reactions is obtained by assigning meaning to the frequencies, instead of directly using purely numerical values. The semantics is discussed next.

3.1 Semantics of Event Frequencies

The overall frequency of events should be a superposition of a few independent frequencies. The following classification – from the highest to the lowest frequency – is the basis of the CyboPlayer semantics:

1. *Background Noise* – this the highest random frequency, upon which other frequencies are added;
2. *Hourly Frequency* – people usually determine their actions by division of the day into periods of a few hours: morning, afternoon, evening and night; for instance, the hourly frequency determines the distribution of meals during the day;
3. *Daily Frequency* – certain activities are performed on a daily basis, say sleeping vs. being awake, going and returning to/from work;
4. *Weekly Frequency* – it is usual to have a weekly rest day, dedicated to leisure and travel.

Each of the above frequencies should be independently randomized, for different kinds of actions/reactions.

Frequencies can be dealt with directly, or by means of setting time intervals.

3.2 Semantic Approach to Friends' Connection

Friendship relations can be given a purely numerical treatment – viz. absolute number

of friends, or friend acquisition rates.

Friendship relations can also be dealt with by semantic means, with specific implications for community structure. One can classify such relations as:

- Family
- Closer friends
- Acquaintances
- Etc.

In this work, a semantic approach to friendship has not been applied yet.

4 CyboPlayer Architecture

In this section we consider the software architecture of the CyboPlayer.

4.1 CyboPlayer Modules

The CyboPlayer is a software package interacting with the Social Network Server.

The internal modules of the CyboPlayer are seen in Fig. 1. A controller triggers actions in the determined frequencies, based upon an internal timer.

Actions as those described in section 2, can be either independent, interactions with other players or logging function for posterior analysis.

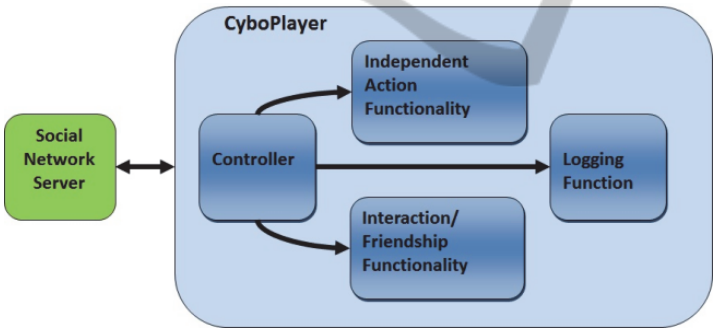

Fig. 1. CyboPlayer Software Architecture – its modules are represented as round (blue) rectangles. The controller has a timer used to trigger actions. These are either independent actions or interactions with friends. For each action performed a log record is saved by the logging function. The whole CyboPlayer interacts with the Social Network Server.

5 Validation

Validation of the hypotheses upon which CyboPlayer is based – the required functionalities and the superposition of randomized frequencies – should be done by analysis of its actions.

The ultimate validity criterion is the long term survival, friendship acquisitions

and natural development of CyboPlayer in the Social Network environment.

5.1 Experimental Technique

Frequencies – on an interval basis – were set based upon statistics published in the literature. For instance, published data per month for an average user are as follows:

- Sends 8 friend requests;
- Clicks Like 9 times;
- Writes 25 comments;
- Is invited to 3 events.

The logging function records all the actions of CyboPlayer in the following format:

Action started – action type, start date;

Action completed – action type, end date;

The log file is able to be read by a software tool that scans records by action types. The tool can generate graphs of actions along the time axis.

5.2 Experimental Results

CyboPlayer has been currently alive, being tested in the laboratory and data being accumulated.

Preliminary results show that it has been performed actions regularly and has acquired friends. Most importantly it has already survived for a few weeks, after initial suffering from infancy illnesses, including unexpected power outages.

CyboPlayer has been interacting in the environment of large commercial social networks.

Detailed results will be published in a longer version of this paper.

6 Discussion

A cyborg has been designed and actually implemented to test its resilience to Turing tests generated by the social network server.

CyboPlayer is currently in its first demonstration tests in the laboratory.

The novel point of view is to look at CyboPlayer as a challenging Turing test, rather than the conventional security perspective, typically represented by say CAPTCHA tools.

Typical robot Turing test competitions focus on chat capability. Our CyboPlayer emphasizes other capabilities, such as reasonable action frequency and reasonable contents of posts. In fact, CyboPlayer does not have yet a chat function.

Technical aspects of programming CyboPlayer are not trivial and demand significant ingenuity. On the other hand, once one has a CyboPlayer alive, one of the most impressive outcomes is how easy it is to deal with the semantic problems to attain the desired resilience.

Interactions with CyboPlayer to a certain extent reminds Eliza by Weizenbaum

[7], an early computer program playing a psychoanalyst role with which people engaged in personal conversations involving quite deep feelings. Eliza had the advantage of a very restricted field and style of conversation.

6.1 Future Work

Beyond the essential validation of the CyboPlayer design and assumptions, there are several issues deserving future investigation.

First, what is the spectrum of algorithms for systematic generation of frequency parameters. More sophisticated algorithms may provide the basis for online learning/adaptation of the frequency parameters.

Another issue is the application of semantics in the determination of friends' acquisition rates and community structures.

Improvement of the content related functions – say posting of comments – and finally reaching chat capability will also be dealt in later stages of this research.

Finally, it should be interesting to generate a few instances of CyboPlayers that will be able to interact among themselves and with human players in the social network.

6.2 Main Contribution

The main contribution of this work is the usage of semantics to determine action frequencies for the designed cyborg, to improve its immunity to Turing tests.

References

1. Boshmaf, Y., Muslukhov, I. Beznosov, K. and Ripeanu, M., "The Socialbot Network: When Bots Socialize for Fame and Money", in Proc. ACSAC'11, 2011.
2. Cha, M., Mislove, A. and Gummadi, K. A., "A measurement-driven analysis of information propagation in the flickr social network", in Proc. 18th International Conference on World Wide Web, 2009.
3. Chu, Z., Gianvecchio, S., Wang, H. and Jajodia, S., "Who is Tweeting on Twitter: Human, Bot or Cyborg?", in Proc. ACSAC'10, pp. 21-30, 2010.
4. Gianvecchio, S., Xie, M., Wu, Z. and Wang, H.. "Measurement and classification of humans and bots in internet chat", In Proc. 17th USENIX Security Symposium, San Jose, CA, 2008.
5. Newman, M. E. J., "Detecting community structure in networks", Eur. Physical Journal, B, Vol. 38, pp. 321–330, 2004.
6. Turing, A. M., "Computing machinery and intelligence", Mind, Vol. 59, pp. 433–460, 1950.
7. Weizembaum, J., Computer Power and Human Reason, Freeman, San Francisco, USA, 1976.
8. Yan, J., 2006. "Bot, cyborg and automated turing test", in Proc. 14th International Workshop on Security Protocols, Cambridge.