The Smart-TSH-Finder: Crawling and Analyzing Tempo-Spatial Hotspots in Second Life

Akram Al-Kouz , Ernesto William De Luca, Jan Clausen, Sahin Albayrak
DAI-Labor, TU Berlin, 10587 Berlin, Ernst-Reuter-Platz
{Akram, ernesto.deloca, Jan.clausen, sahin.albayrak}@dai-labor.de

Abstract

In this paper we introduce the Smart Tempo-Spatial Hotspots Finder (Smart-TSH-Finder) for smart data crawling in the Second Life (SL) virtual world. Classical methods of crawling data from SL lead to irrelevant data content because of the dynamic nature of avatars and objects in SL. In order to build artificially intelligent expert avatar agents that are able to provide intelligent services for other typical avatars in virtual world we attempt to enhance the quality of extracted data from SL. Based on experimental observation, avatars tend to gather in some places for different amounts of time, which forms temporal and spatial hotspots. Utilizing the Tempo-Spatial characteristics of the avatars behavior in virtual worlds could improve the quality of the extracted data. Smart-TSH-Finder implements a Tempo-Spatial Hotspots finding mechanism to crawl dynamic contents such as chat conversations from Second Life. The system introduces two mechanisms: the Tempo-Spatial Hotspots Detection, and the Tempo-Spatial Hotspots Prediction. Our smart chat conversations that have been crawled showed good enhancement in content quality of the crawled chat conversation which will enrich the future textual analysis work. Additionally, we found that extracting avatars interactions and behavior in the Tempo-Spatial Hotspots in addition to chat conversations can help in generation a more coherent social network model for SL.

1 Introduction

Second Life[1] (SL) is a 3D virtual world developed by Linden Labs[2], a free client program called the Viewer enables its users to interact with each other through avatars. Avatars can explore, meet, socialize, participate in individual and group activities. SL has monthly unique users with repeat logins pick of 826,214 in March 2010; this represents 7% growth over the high set in Q4 2009, and 13% growth over March 2009[3]. The increasing popularity of SL as immersive virtual world made it an interesting target of research. In SL we have observed that avatars tend to gather in certain places (hotspots) in the land because of their sense to persistency. According to the operative definition of presence of [Vives and Slater, 2005], in that we suppose avatars will behave in the same way in SL. In order to build smart expert avatar agents able to provide smart services for other typical avatars in virtual world we need to extract dynamic data contents from SL. A data crawler capable to extract the desired data is needed. But SL is a commercial product, we do not have access to statistical data, and it is unlikely that such data would be made public by Linden Labs. On the other hand SL puts restrictions on the amount and the nature of data that can be crawled. The land in SL is divided into Parcels which are subdivided in a 256x256 meter region, content is only presented to users as they move close to the content location, thus, user can extract the public chat log within the silence cone of 20 meters, added to his own behavior and interactions with objects in SL, so the crawler must dynamically model the environment based on the previously mentioned restrictions.

We analyzed and utilized the Tempo-Spatial characteristics of the avatars behavior in SL to build Tempo-Spatial Hotspots finding system which contains smart agents (bots) able to extract dynamic contents such as chat conversations from intelligently detected or predicted hotspots in SL. Our goal was to determine trends in SL by analyzing the content of the extracted chat logs in future work to discover behavior patterns, conversation patterns, and conversation topics and to structure a social network in SL. Furthermore, evaluating how useful this information is in building user profile to be used in future work for recommendation or integration in our previous work done with the SpreeLand Expert Avatar [Al-Kouz et al., 2010]. We developed the proposed system based on the LibOpenMetaverse[4] open-source software related to the metaverse and virtual worlds. While our system is compatible with SL, it is also capable to integrate with other alternative virtual worlds in OpenSimulator[5].

In this work, we first present related work in the next section. Then, we introduce our system architecture and the related analysis mechanisms and algorithms used to build it, in the following section we discuss the empirical evaluation experiment. The paper concludes with a discussion and future work.

[1] www.secondlife.com
[2] http://lindenlab.com
[3] http://blogs.secondlife.com/community/blog/2010/04/28/economy-hits-new-all-time-high-in-q1-2010
[4] LibOpenMetaverse http://www.libsecondlife.org
[5] http://opensimulator.org/wiki/Main_Page

2 Related Works

At the moment, there is no previous work to smart use of the Tempo-Spatial characteristics of the avatars behavior in SL and related virtual worlds for enhancing the quality of crawled data. However, there are some published studies about utilizing both of the characteristics separately. Due to the lack of publicly chat data sets and the closed nature of SL as commercial product, the field has not yet gained great interest as opposed to the currently prevailing research on the typical Instant Messaging. Obviously, some research presents an intelligent agent crawler designed to collect user-generated content in SL. The agents navigate autonomously through the world to discover regions. It shows that virtual worlds can be effectively extracted using autonomous agent crawlers that emulate normal user behavior [Eno et al., 2009]. A refined approach presented analysis and visualization of the spatial and temporal distribution of user interaction data collected in 3D virtual worlds [Börner et al., 2001]. However it is not discussing its effect on the extracted data quality. A more sophisticated model developed a chat message analysis system called IMAnalysis using text mining techniques, which includes functions for chat message retrieval in general and not dedicated to 3D virtual worlds [Hui et al., 2008]. There has been some work on crawling and exploring virtual worlds for reasons other than content collection [Varvello et al., 2008], [La and Pietro, 2008]. Even though, none of mentioned researches used Tempo-Spatial characteristics of SL to enhance the quality of crawled data.

3 Our Approach

Based on our observation, SL avatars tend to gather in specific trendy spots for variant amount of time, taking this fact into consideration adding to the constraints on data crawling by Linden Labs, we find that the classical methods of crawling data from SL such as extracting data by a logged in automated bot into SL in specific location, or by making the bot to follow some avatars to imitate the natural behavior of the avatars, lead to irrelevant data content, because of the dynamic nature of avatars and objects in SL. In our attempt to enhance the quality of extracted data from SL, a smart data crawler system based on the Tempo-Spatial analysis of the avatars behavior was developed to crawl dynamic contents such as chat conversations from SL. Our system is based on the client/server protocol of the open source LibOpenMetaverse library.

3.1 System Architecture

As we have shown in Figure 1, the architecture of our proposed Smart-TSH-Finder system has two main modules: the Virtual Environment Layer module and the Smart-TSH-Finder Crawler module. The first establishes the network communications with the virtual world. The last contains three sub modules: Login Manager, Heat Maps Manager, and Crawling Manager. Heat Maps Manager introduces two alternative smart mechanisms to find the trendiest hotspots in the virtual world suitable to crawl dynamic contents from: the Tempo-Spatial Hot Spots Detection service, and the Tempo-Spatial Hot Spots Prediction service.

The Virtual Environment Layer Module

The Virtual Environment Layer Module is an intermediate network abstraction layer between Smart-TSH-Finder Crawler and the virtual world, it was built on top of LibOpenMetaverse library to customize some of the functionalities required by our Smart-TSH-Finder Crawler. LibOpenMetaverse runs on top of the .NET runtime and maintains compatibility with the Second Life protocol and can be used for creating clients and automatons in Second Life, OpenSimulator or other virtual worlds which use the Second Life Protocol.

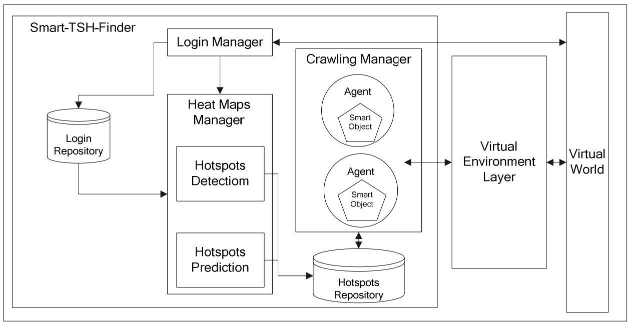

Figure1: Smart-TSH-Finder Architecture

The Smart-TSH-Finder Module

The Smart-TSH-Finder Module is a distributed artificial intelligence agent built on top the Virtual Environment. It is capable to connect multiple lands and extract dynamic contents such as chat conversations logs and some avatars' behavior actions. Smart-TSH-Finder Crawler has three sub modules: the Login Manager sub module, the Heat Maps Manager sub module and the Crawling Manager sub module.

The Login Manager Sub Module

The Login Manager is capable to connect to multiple lands simultaneously, it handles the authentication processes with the virtual environment, an active SL user credentials are required. After login it stores all the standard retrieved data from the virtual world in the Login Repository as explained in Figure 1.

The Heat Maps Manager Sub Module

The Heat Maps Manager is the core of our system, it uses the data stored in the Login Repository to generate a spatial heat map for the currently logged in land as shown in Figure 2, taking into consideration the virtual world specifications, such as the size of the Parcel (256*256 meters) and silence cone (20 meters), it generates a 2D matrix corresponding to the spatially discovered hotspots. The generated 2D matrix is used as the base for the smart Tempo-Spatial analysis in the Detection Service and the Prediction Service to determine the trendiest spots that are suitable to extract informative data from.

Detection Service

The Detection Service implements the Detection Algorithm which will be discussed in the first part of the Smart-TSH-Finder Algorithms section to discover the trendy Tempo-Spatial spots. It will suggest the most fitting spots between temporal and spatial spots, added to that it has the capability to use a baized criteria specified by the user to retrieve hotspots temporally or spatially baized. The retrieved hotspots will be stored in the Hotspots Repository.

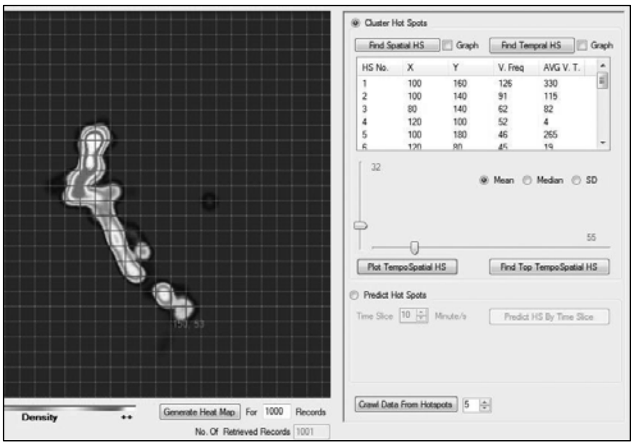

Figure 2: Heat Maps Manager Screenshot.

Prediction Service

The Prediction Service implements the Prediction Algorithm which will be discussed in the second part of the Smart-TSH-Finder Algorithms section to predict the next trendy spatial spots based on time stamp. It will predict the trendiest spots based on a user selected time slice, and predict the next probable hotspot using linear regression. The predicted hotspots will be sorted in the Hotspots Repository.

Crawling Manager Sub Module

The Crawling Manager is responsible for management of the data crawling process. It uses the data stored in the Hotspots Repository to generate a smart agent for each hotspot, the generated smart agent is programmatically controlled bot, which has the capability to extract dynamic contents from the virtual world. Furthermore, it can handle some Smart Objects. Smart Object is a geometric object, such as cube or circle, manually prebuilt in a free land and attached to the user's avatar, it has script able to response to some interactions and extracts some data. The extracted data will be stored in the Hotspots Repository for further analysis.

3.2 Smart-TSH-Finder Algorithms

In this section we describe the algorithms used to analyze the Tempo-Spatial behavior of the avatars based on the spatial heat map for the currently logged in land generated by the Heat Maps Manager as shown in Figure 2. The corresponding 2D matrix will be used as the base for the smart Tempo-Spatial analysis mentioned in the Detection mechanism and the prediction mechanism to determine the hotspots that are suitable to extract data from.

Hotspots Detection Algorithm

In the following we present an algorithm by which we detect hotspots of the land. The detection algorithm is an unsupervised learning method to assign set of Tempo-Spatial observations which are the visits frequency of the location as spatial factor and the average visit time of the location as temporal factor. In our experiment of the discovered temporal and spatial hotspots exhibit a Pearson correlation coefficient of approximately 0.74, details are provided by Figure 3, where the x axis is a numeric representation of the hotspot, the left y axis represents the visits frequency and the right y axis represents the average visit time in second, the plotted curve with diamonds represents the spatial values and the curve with circles represent the temporal values.

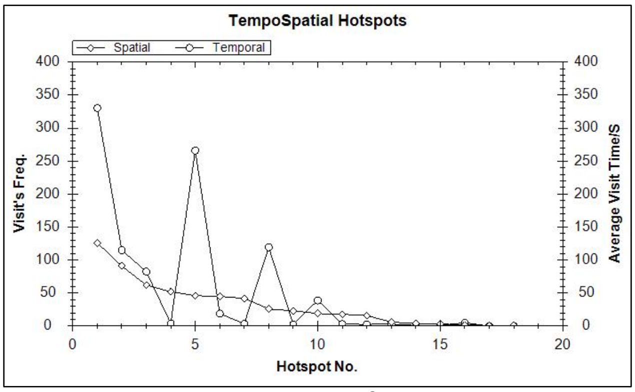

Figure 3: Tempo-Spatial Relation of Hotspots.

Following we describe how Tempo-Spatial Hotspots are discovered. First the mean or median respectively for each coordinate is computed. If a spot has a value greater than the mean in each component, it is considered as a Tempo-Spatial Hotspot. By this we find the most suitable fitting between the temporal and spatial hotspots. Furthermore, the Detection Service has alternative statistical techniques to find fitting between the temporal and spatial hotspots leading to discovering the most trendy hotspots both temporally and spatially as shown in Figure 4, where the x axis is the average visit time per hotspot and the y axis is the visits frequency of that hotspot, the fitted hotspots are plotted as big circles, the others as small diamonds.

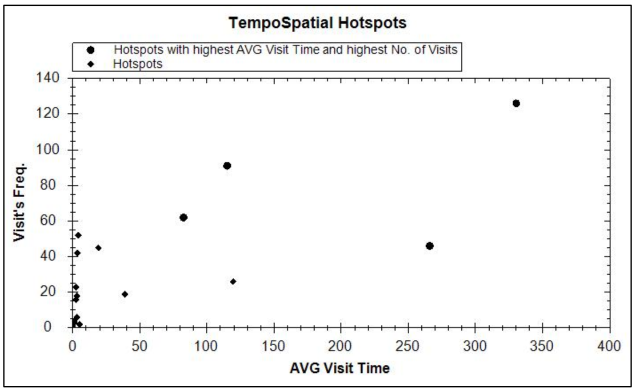

Figure 4: Tempo-Spatial Fitted Hotspots.

The default fitting criteria is the logical AND between the mean of the temporal hotspots and the mean of spatial hotspots, further the user can alternate between median and standard deviation as fitting criteria, fitted hotspots should be greater than the components means, median, or standard deviation respectively. On the other hand, the Detection Service will enable the user of the system to specify the biases into the temporal direction to crawl long chat conversations with low number of participants, or into the spatial direction to crawl long chat conversations with high number of participants.

Hotspots Prediction Algorithm

The Hotspots Prediction Algorithm uses the previously generated Heat Map to predict the coming hotspots in terms of time. It uses the linear regression analysis to expect the most top hotspots and its sequence in the next time slice. The spatial hotspots will be classified based on the user selected time slice generating a 2D matrix HS [i, j]. The first index i is a representation of the hotspot number, the second index j is a representation of the accumulated time slices and the value at HS [i, j] is the visit frequency of the ith hotspot at the jth time slice as shown in the Table 5.

HS/Time	*i*1	*i*2	*i*3	*i*4	*i*5	*i*6	*i*7
*j*1	124	24	75	2	4	133	41
*j*2	93	53	55	4	3	30	8
*j*3	24	45	2	3	12	33	17
*j*4	53	13	12	0	0	18	0
*j*5	0	10	22	0	5	10	20
*j*6	0	0	0	0	12	12	0
*j*7	0	0	5	0	0	0	12

Table 5: Sample Hotspots Matrix

The Hotspots Prediction Service generates a 3D surface representation of HS matrix, as shown in Figure 6 hotspots reach their peak at different times. The system use the current time stamp to predict the next trendy spot by retrieving the correspondence visit frequency of all the hotspots in that time interval into two vectors, one to store the hotspot number and the other to store the corresponding visit frequency. The Linear Regression algorithm will be applied on the two vectors, hotspot with the least absolute value of residuals will be suggested as the next hotspot.

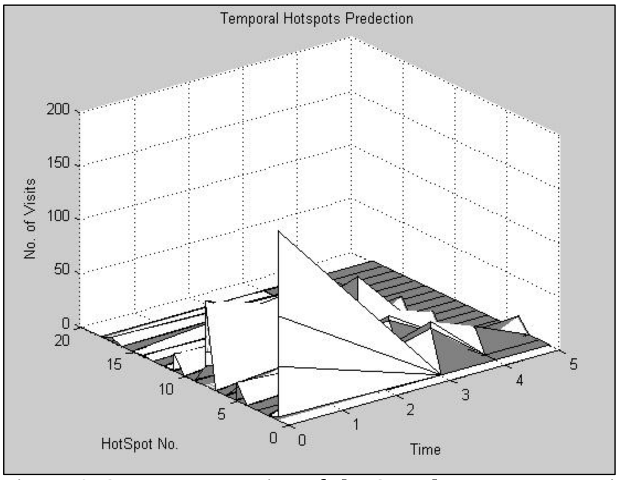

Figure 6: 3D Representation of the Sample Hotspots Matrix in Table 4.

4 Evaluation

In order to evaluate our system, we compare the quality of the extracted data in both the classical methods and in our proposed method based on the mentioned Tempo-Spatial Hotspots Finding Algorithms.

4.1 Data Crawling

We used Solace Cove Land[6] to do the experiment, Solace Cove is a popular Second Life virtual space for meeting friends, doing business, and sharing knowledge. We ran our system in the mentioned land five times for thirty minutes every time. Our system found the Tempo-Spatial Hotspots and sent four smart agents to that spots to start data extraction. At the same time we logged a bot in Solace Cove Land using classical LibObenMetaverse dashboard client and start extracting data by enforcing the bot to follow a specific avatar and record the chat log. In the previously mentioned two cases we fired up a specific discussion question "How was the FIFA world cup 2010 in South Africa" several times, and then we extracted avatars responses for thirty minutes. We got four chat log files from our system prefixes with hotspot number each one contains a different number of utterances and one chat log file from the classical dashboard prefixes with "DB" as shown in Table 7.

4.2 Data Preparation

In order to compare the quality of the crawled data from our system with the quality of the crawled data from the classical dashboard, we applied a windowing mechanism to split chat log files into smaller files. We considered the window size of eight utterances, six utterances as the average size of chat conversation based on [Ogura *et al.*, 2004], one for pre overlapping and one for post overlapping. Utterances fit in each window are stored as a separate file prefixes with chat log file name and window number as in Table 7.

Chat Log Files	No. of Utterances	No. Of Windows
HS1ChatLog	213	35
HS2ChatLog	322	53
HS3ChatLog	105	17
HS4ChatLog	96	16
DBChatLog	386	64

Table 7: Sample Extracted Chat Log files and its Windows

4.3 Evaluation Process

In our evaluation process we considered the previously fired question "How was the FIFA world cup 2010 in South Africa" as query, and we used a ranking function based on summing the TF–IDF[7] weight (term frequency–inverse document frequency) for each query term to evaluate how important that term is to a document in a collection or corpus. In this case each window file considered as a document and all the windows corresponding to one chat log file considered as corpus. To normalize the results we evaluated the first windows as the number of windows in chat log file of the lowest number of windows. We used the algorithm in Figure 8 to rank each chat log file. Chat log files with higher rank are the most relevant to our query.

4.4 Evaluation Results

We conducted five runs of our experiment and took the average results to generalize observations. Average results show that three out of the four hotspots chat log files (HS1ChatLog, HS3ChatLog and HS4ChatLog) which were extracted by our system have higher rank than the dashboard chat log file "DBChatLog" which extracted by the classical method. Thus, chat log file "HS2ChatLog"

[6] http://maps.secondlife.com/secondlife/Solace20Cove/121/81/24

[7] http://nlp.stanford.edu/IR-book/html/htmledition/tf-idf-weighting-1.html

has rank lass than but close to the rank of dashboard chat log file "DBChatLog". Further analysis of the results and extracted data has been done to discover that dashboard chat log file "DBChatLog" beyond chat log file "HS2ChatLog" in rank because the followed avatar used in the classical way hanged in one place for long time and that place was selected by our system as a hotspot, so our extracted data at this hotspot is a subset of the "DBChat-Log" file which explains why we got low rank in this hot-spot. Beyond that our system showed high ranking for the extracted data from the discovered hotspots which means our extracted data is more relevant and coherent.

For each chat log file
  For each window
    ChatLogFileRank += *WindowRank*

$$WindowRank = \sum_{iTerm}^{nTerm} TF \times IDF$$

Terms= "word", "cup", "2010","south","africa"
iTerm: First entry in Terms
nTerm: Last entry in Terms

$$TF = \frac{TT}{TW}$$

TT: No. of times term appears in window
TW: No. of words in window

$$IDF = Log\left(\frac{W}{WT}\right)$$

W: No. of windows in chat log file
WT: No. of the windows the term appeared in

Figure 8: Chat Log File Ranking Algorithm

5 Conclusion and Future Work

A Smart-TSH-Finder for Second Life has been developed as a substitute to the classical methods of crawling data from SL. The proposed system enhance the quality of the extracted data, which considered a major step in building artificially intelligent expert avatar agents able to provide smart services for other typical avatars in virtual world. Smart-TSH-Finder Crawler utilized the Tempo-Spatial characteristics of the avatars behavior in virtual worlds to improve the quality of the extracted data to discover the temporal and spatial hotspots that avatars tend to gather in. The system introduces two mechanisms: the Tempo-Spatial Hotspots Detection, and the Tempo-Spatial Hot-spots Prediction. System user have the choice to use the former or the last mechanism to determine the most trendy hotspots, after that the system creates an agent for each trendy hotspot and send it to that hotspot, the generated agent starts crawling data from its desired hotspot.

The empirical evaluation experiment of the crawled chat conversations showed good enhancement in content quality which enriches the future textual analysis work to discover behavioral patterns, conversation patterns, and conversation topics. Additionally, we found that extract-ing avatars interactions and behavior in the Tempo-Spatial Hotspots in addition to chat conversations can help in modeling a social network for Second Life. Furthermore, evaluating how useful this information is in building user profile to be used in future work for recommendation or integration with our previous work done with our previous work.

References

[Vives and Slater, 2005] Sanchez Vives, M.V., Slater, M, From presence to consciousness through virtual reality. Nature Reviews Neuroscience 6(4), 332–339, 2005.

[Al-Kouz *et al.*, 2010] Akram Al-Kouz, Winfried Um-brath, Ernesto William De Luca, Sahin Albayrak, SpreeLand: An Expert Avatar Solution for Virtual World E-Learning Environments, the International Conference on E-Learning in the Workplace, 2010.

[Eno *et al.*, 2009] Joshua Eno, Susan Gauch, Craig Thompson, Intelligent Crawling in Virtual Worlds IEEE/WIC/ACM International Conference on Web In-telligence and Intelligent Agent Technology – Work-shops, 2009.

[Börner *et al.*, 2001] Katy Börner, William R. Hazlewood & Sy-Miaw Lin, Visualizing the Spatial and Temporal Distribution of User Interaction Data Collected in Three-Dimensional Virtual Worlds, 2001.

[Hui *et al.*, 2008] Hui, S.C. , Yulan He , Haichao Dong , Text Mining for Chat Message Analysis, IEEE Confer-ence on Cybernetics and Intelligent Systems, 2008.

[Varvello *et al.*, 2008] M. Varvello, F. Picconi, C. Diot, and E. Bier sack, Is there life in Second Life?, Thom-son Technical Report CR-PRL-2008-07-0002, 2008.

[La and Pietro, 2008] C-A. La and M. Pietro, Characteriz-ing user mobility in Second Life, Technical Report RR-08-212, Institut Eurecom, 2008.

[Ogura *et al.*, 2004] Kanayo Ogura, Masato Ishizaki and Kazushi Nishimoto, A Method of Extracting Topic Threads Towards Facilitating Knowledge Creation in Chat Conversations, Knowledge-Based Intelligent In-formation and Engineering Systems, 2004.