Modeling and Analyzing the Video Game Live-Streaming Community

Gustavo Nascimento*, Manoel Ribeiro*, Loïc Cerf*, Natália Cesário*, Mehdi Kaytoue†,
Chedy Raïssi‡, Thiago Vasconcelos* and Wagner Meira Jr.*
* *Universidade Federal de Minas Gerais, Departamento de Ciência da Computação, Belo Horizonte, Brazil,*
Email: {gunasper, manoelribeiro, lcerf, natalia, thiagorova, meira}@dcc.ufmg.br
† *Université de Lyon, CNRS, INSA-Lyon, LIRIS, UMR5205, F-69621, France,*
Email: mehdi.kaytoue@insa-lyon.fr
‡ *INRIA Nancy Grand Est, 54500, Vandœuvre-lès-Nancy, France,*
Email: chedy.raissi@inria.fr

***Abstract*—In parallel to the exponential growth of the gaming industry, video game live-streaming is rising as a major form of online entertainment. Gathering a heterogeneous community, the popularity of this new media led to the creation of web services just for streaming video games, such as Twitch.tv.**

In this paper, we propose a model to characterize how streamers and spectators behave, based on their possible actions in Twitch and, using it, we perform a case study on the Starcraft II streamers and spectators. In the case study we analyze a large amount of data collected in Twitch.tv's chat in order to better understand how streamers behave, and how this new form of online enteirtainment is different from previous ones.

Based on this analysis, we were able to better understand channel switching, channel surfing, and to create a model for predicting the number of chat messages based on the number of spectators. We were also able to describe behavioral patterns, such as the mass evasion of spectators before the end of a streaming section in a channel.

***Keywords*-video game; starcraft; streaming; twitch.tv;**

I. Introduction

Over the last decade, the annual turnovers generated by the electronic entertainment industry went beyond those of both cinema and music industries, making video game production a highly profitable business. A recent example is the announcement of a budget of US$ 500 millions for the video game Destiny (Blizzard/Activision) to be released in September 2014[1] eclipsing the previous record, Grand Theft Auto V (Rockstar Games), whose budget was US$ 265 millions and had an realease date earning of US$ 800 millions[2]. As a matter of fact, video games are challenging Hollywood blockbusters.

In parallel with the game industry exponential growth, a very related yet totally different form of online entertainment is also expanding at a very fast pace. Watching video-game live-streams is becoming an increasingly popular way of entertainment. From casual players recording themselves playing indie games to highly competitive e-sports international championships, the content produced has attracted a huge and heterogeneous community all over the world [1]. This may sound very peculiar at first glance as video games are primarily designed for the players, but so were designed physical sports with millions of weekly spectators, such as football or basketball.

This scenario has led to the creation of live-stream web services just for video games, like Twitch.tv. Started in 2011 from a Justin.tv fork, Twitch has grown quickly and met a huge success [2], [3]. In 2014, Twitch counts more than 45 million unique viewers per month gathered around for broadcasting, watching, and chatting from everywhere they play. The web service holds a very diverse set of sub-communities such as the *e-sports*, focused on highly skilled players and huge championships [4], the *speed running*, where the main objective is to finish a game as quickly as possible (often, beat a world record), and the *let's play*, characterized by charismatic streamers that interact a lot with his or her spectators [5]. Twitch is also used to stream (and promote) beta games, plays the role of targeted advertising of all kinds of games, and was included in several games consoles, such as PlayStation 4 and X-Box One systems. In February 2014, Twitch was 4^{th} in Peak US Internet Traffic, being responsible for 1.8% of the traffic, just behind Netflix, Google and Apple [6]. Understanding who produces and consumes this important volume of video content is as such a major interest. In summary, there is as clear a trend towards streaming ones games, as there is for one tweeting about his life.

In 2010, Kaytoue et al. made a first characterization of the e-sport live-streaming sub-community within Twitch [1]. In this paper we propose a model to analyze how the streamers and spectators behave and further characterizing the live-streaming community through a case study. The model consists of mapping the actions that spectators and streamers can perform as transition graphs, enabling us to analyze qualitatively and quantitatively the video game

[1]http://en.wikipedia.org/wiki/Destiny_%28video_game%29
[2]http://en.wikipedia.org/wiki/Grand_Theft_Auto_V

978-1-4799-6953-1/14 $31.00 © 2014 IEEE
DOI 10.1109/LAWeb.2014.9

live-streaming community. Using the proposed model, we are able to answer three fundamental questions about the community: (i) How do streamers and spectators behave? (ii) Are there patterns in those behaviors? and (iii) How is the content in the video game live-streaming community different from other kinds of online entertainment?

In our case study, we analyze the Starcraft II streamers and spectators, clearly part of the e-sports sub-community. Already studied in other papers [1], [7] and books [4], Electronic Sports started to really develop in the 90's with licenses like Doom and Counter Strike. Starcraft Broodwar (Blizzard Entertainment) was a huge success in South Korea: competitions were even casted on TV on prime time. Its successor, Starcraft II, met similar success, having its own world-wide player ranking system (ELO) and annual world cup competition series (WCS) with a US$1,6 million prize pool for the year 2014. Electronic sports would not be so developed if they were not supported by strong and active communities around the world. Reaching such communities is not possible though classic media. On the other hand, the usage of Social TV, or live streaming, works as a catalyst, that is, a mechanism to meet, discuss, and share the passion about e-sport.

Analyzing the data related to this fascinating environment, we discovered a lot about the behavior of both streamers and spectators. For instance, we were able to: (i) understand the meaning of switching from one streamer channel to another, (ii) describe the mass evasion of spectators that happens in the last minutes before and after the end of a streaming session, (iii) identify channel surfing as a behavioral pattern, and (iv) create a model to predict the amount of chat based on the number of spectators logged into a channel. We believe that these results are of major interest for the whole gaming community, and a large step into fully understanding this new forms of entertainment that are e-sports and video game live-streaming.

The rest of this paper is organized as follows. The second section discusses related works. In the third section we present some background concepts to the video game live-streaming context. In the fourth section we present our model for the video-game live-streaming community. In the fifth section we describe a case study with the Starcraft II streamers' channels, trying to answer the fundamental questions mentioned before. Finally, in the sixth and last section we conclude the paper.

II. Related Work

A first analysis of live streaming workloads on the internet can be found in [8]. We are more specifically interested here in the so called social TVs, which combine communication and social interactions in a TV framework [9], [10]. As the video game live-streaming is a kind of social TV, allowing interaction of the users via chat, the subjects are rather related. In [11] the social television model is described and analyzed, providing a framework to understand the current situation of social TV and identifying future developments. Similarly, [12] describes empirical results of computer-mediated groups using social TV, characterizes its model and suggests features to future social TV prototypes. Despite being closer to the traditional television content, both works provide valuable information about how the spectators interact in social TV, which can be applied to the video game live-streaming scenario.

Video games were also studied in the literature. In fact, not only the games themselves but how the players play those games, which was the most frequent issue. Despite differences from our work in this paper, [13] discusses how players learn to be grandmasters in games like Starcraft II and chess. They observed the entire training process and interviewed actual players in order to grasp the main aspects that lead a novice player to become a grandmaster. Whereas there are plausible cognitive markers of expertise that can be identified from the games logs (recording all players actions) [14], we may link the work of [13] and of [7] to find out that there are many novice players who learn by watching live stream games from grandmasters.

Finally, there are some works about both e-sports and video game live-streaming. [4] discusses multiple facets of the e-sports scene: pro-gaming, its highly paid players, play-by-play broadcasts, and mass audience, it also describes the whole e-sports environment of leagues, teams, organizers, sponsors and fans. In [1] we made a first characterization (based on collected data) on the e-sport community. In [7] the nature of the Starcraft II spectator (and from spectatorship itself) is studied; the paper proposes personas that represent the different reasons for people watching gamers playing, and the relevance of such habit to the larger video game live-streaming context. In [5], a broader view of the video game live-streaming context is presented. The work discusses the web services for live-streaming and describes the major sub-communities.

Our article has significant differences from the aforementioned related work. We perform a significant data analysis, in contrast with [4], [5], [7], which are quite theoretical, while proposing a model that captures the semantics of streamers and spectators interaction, going beyond data-driven analysis [1]. Further, we focus on the e-sports sub-community as [1], [4] but the model we propose is generic and may be used to analyze any e-sports sub-community. Our paper is also different from those about social TVs and from the gaming content, because the fusion of those two elements creates a completely different scenario which is the video game live-streaming.

III. Video game live-streaming

Before introducing our characterization model, we need to present, in this section, some background on video game live-streaming.

Twitch.tv is a platform that provides channels where users stream themselves playing and other users may watch them. The website also allows users to chat in real time in a given channel as well as provides an API that can be used to gather data about the streams. Spectators may choose channels using a search engine, browse the featured channels, or view channels by game or broadcaster. Twitch.tv makes money by placing advertisements over streamed content and by featuring sponsored channels, but it also allows streamers to monetize their streams. Spectators may, for example, subscribe to the channel of a given streamer to support his or her work.

The streamer is the user who streams live content on his or her channel—a web page in Twitch. The streamer may perform three actions in Twitch: (i) start a stream, (ii) prepare to end a stream, (iii) end a stream. Notice that preparing to end the stream is when the streamer signals that he or she will end the stream. In those moments the streamer bids farewell to the spectators and makes acknowledgements. We decided to highlight this action because we observed that it is usual that a significant number of spectators leave the channel when the streamer prepares to end the stream. The majority of the streams are generated by single persons, but teams or even championships are often streamers. An important observation is that, in Twitch.tv, each streamer has a channel where spectators may join regardless of the streaming being active.

The chat is an important part of the video game live-streaming experience because it enables both the spectator/spectator and the streamer/spectator interactions. The conversation subjects in the chat vary significantly, being very dependent on its associated sub-community, ranging from comments about the streamer performance to strategies about the game being played [5].

We define session as the time interval between the moment when the streamer starts to stream and the moment he or she ends the stream. Many streamers maintain a schedule of his or her sessions so that spectators know when his or her channel is online.

The spectator is the user who watches streams and may perform 4 actions in Twitch: (i) join a channel, (ii) leave a channel, (iii) send messages in the chat of a channel, and (iv) switch from one channel to another. The last action may be seen as a joint action that consists of leaving the channel of a streamer X and joining the channel of a different streamer Y. In [7] several stereotypes of video game live-streaming spectators were described. Examples of those stereotypes are people who watch professional players to improve their own gaming techniques and people who enjoy watching other people playing more than actually playing the game. Distinct spectators watch gaming live streams for distinct reasons.

IV. Our model

In this section we propose a model to characterize how streamers and spectators behave based on their actions in Twitch. In particular, we propose a transition graph for each role based on the actions described in Section III. The vertices in the transition graphs are states and the transitions are actions.

In our model, a streamer α is in one of three possible states: (i) streaming (ON), (ii) disconnecting (OD), that is, preparing to end the stream, and (iii) not streaming (OFF). We may then define the actions that α may perform as transitions between states:

Action	Src.	Dest.
Start a stream α	OFF	ON
Prepare to end a stream α ($\sim 3m$ before)	ON	OD
After ending a stream α ($\sim 15m$ after)	OD	OFF

For modeling a spectator S, we should take into account that he or she may have joined multiple channels, so there is a transition graph S_α for each pair streamer and spectator. S_α has two states: (i) connected (IN) and (ii) disconnected (OUT), depending on whether or not S has joined the channel. Next we present the actions of S and respective transitions:

Action	Src.	Dest.
Join channel	OUT	IN
Leave channel	IN	OUT
Chat	IN	IN

The transition graphs for both streamers and spectators are depicted in Figures 2 and 1, respectively.

Using this model, we may record the dynamics of Twitch.tv as a sequence of actions, where each action is a tuple $< G, s, a, d, t >$, where G is a transition graph (associated with either streamer or pair streamer-spectator), s is the source state, a is an action, d is the destination state, and t is a time-stamp.

For instance, we may easily represent when a spectator S switches from channel α to channel β through the transitions $< S_\alpha, IN, leavechannel, OUT, t_1 >$ and $< S_\beta, OUT, joinchannel, IN, t_2 >$. In this case, we may also define that a channel switch must happen within a time interval thr, which is easily formalized by the constraint $|t_1 - t_2| \leq thr$.

The model also allows us to analyze using data from both transition graphs, analyzing the states of the streamers involved when a spectator performs a set of actions. Using this, we may add another level of depth to the switch representation. We define an $A \rightarrow B$ switch as the set of transitions previously mentioned where the spectator leaves a channel where the streamer is in the A state, and joins a channel where the streamer is in the B state.

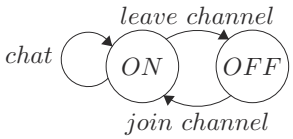


Figure 1: Spectator finite state machine.

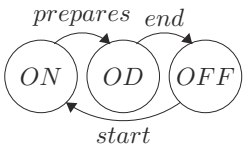


Figure 2: Streamer finite state machine.

We may also use this same idea to analyze the time period during which a spectator joins and leaves a streamer channel while considering the states the streamer was in both moments. We define a sojourn $A \approx B$ as the period of time between a spectator S joins α in state A, and leaves α in state B. Formally, given the sequence of transitions performed by S_α, where the ith transition is join channel, and the $i+1$th transition is leave channel, that is, $< S_\alpha, OUT, joinchannel, IN, t_i >$ and $< S_\alpha, IN, leavechannel, OUT, t_{i+1} >$, we state that $A \approx B = t_{i+1} - t_i$.

V. Starcraft II: a case study

Despite the fact that different games are associated with different workloads and behaviors on Twitch.tv, we believe that it is valuable to characterize and understand in more depth the community behind one specific game, since such understanding would help to grasp invariants that may help characterize other communities as well. We chose Starcraft II, a real-time strategy video game (RTS) with a large community, which has already been analyzed in the litterature [1], [7].

Starcraft II is the fastest selling real-time strategy game of all time.[3] It takes place in a science fiction environment. In that game, every player leads a so-called "race" (among three: Zergs, Protoss or Terrans), exploits resources located on the map (to create buildings, military units and get technological upgrades) and fight the opponents. The player must destroy the buildings of all the enemies to win the game. The victory requires mastering both sophisticated strategies (in an uncertain and real time environment) and a rapid low-level management of the units.

When choosing Starcraft II for our case study, we are actually taking two decisions: (i) to gather data from a single game, and (ii) choosing Starcraft II to be that game. The first decision reduce our sample space but diminishes the noise that gathering data from multiple games would implicate, due to the heterogeneity of Twitch sub-communities [7]. The second decision was taken because Starcraft II was already analyzed in other papers [1], [7], and because it is a game with a solid spectator community.

[3]http://www.eurogamer.net/articles/2010-09-01-starcraft-ii-sells-3-million-in-a-month

spectator	spectator single id.
channel	streamer single id.
date, time	the date and time the tuple was collected.
action	action performed by the user; the action can be joining a channel (INC), leaving a channel (OUT), or sending a message in the chat (CHAT).
chat	the message sent on the chat if the action is CHAT; empty for INC or OUT.

Table I: Elements in a tuple.

Collection period:	**02/10/13** to **17/02/14**
#spectators:	$1,460,740$
#channels:	136
#INCs:	$20,188,434$
#OUT:	$20,195,358$
#CHAT:	$13,878,122$
#tuples:	$54,247,484$
#sessions:	$4,944$
average session duration:	5,2h
median session duration	3,7h

Table II: General information about the dataset.

A. Dataset

Our analysis focuses on the Starcraft II players who reached, at some point, more than one thousand spectators, and the spectators watching those players. The REST API of Twitch.tv allowed us to identify those popular players. The dataset consists of the states of the streams (as given by the REST API) along 139 days and the logs of the chats associated with those streams. An IRC client, written in Python, collected the chat data. Table II summarizes our dataset. It is a set of $54,246,484$ tuples in the format $< user, channel, date, time, action, chat >$, as detailed in Table I.

Importantly, the dataset contains only registered users, i.e., users with an account on Twitch.tv. The remaining spectators cannot chat and no information on them is available but their total number on every channel.

B. Pre-processing the data

The rough data, for a given streamer, suffer from a few inconsistencies: some CHAT actions before the INC actions of the related users, some INC actions of users who are already connected (i.e., several INCs without OUTs in between), some OUT actions of users who are already disconnected (i.e., several OUTs without INCs in between) and users who appear connected but are inactive and never disconect.

The inconsistencies seem to occur because of the IRC's delay when delivering messages, once we consider the time when messages arrive and not when they're sent. On another hand, we also got some errors when our client is rebooting and some OUT actions are lost. The last scenario occurs 2 times a day and its duration is about 2 minutes.

The inconsistencies above affect less than 1% of the data. However, we decided to solve them, applying the following fixes (in the reported order):

1) whenever an apparently offline user chats, add an INC action just before (same time) its first CHAT action;
2) whenever there are several INC actions that occur in less than 1 minute without an OUT action in between, only keep the first one (notice that the fix at step 1 frequently creates this issue, once the real INC action uses the appears some seconds late);
3) whenever there are several INC actions that occur in more than 1 minute without an OUT action in between, add an OUT action after (same time) the last CHAT action. Notice that if she didn't write anything, nothing occurs;
4) whenever there are OUT actions without an INC action in between, or INC actions without an OUT action in between, remove the action.

C. The OD State

By definition, the OD state starts little before the disconnection of the streamer and ends little after that disconnection. However how "little" before and after? We decided to answer this question by analyzing the data. We assume that the OD state coincides with the spectators leaving the channel (hence the chat) at a higher rate.

For every session, we map the maximum number of users the session reached to 1 and proportionally compute a number between 0 and 1 for every other timestamp. After this per-session normalization, the numbers for all sessions (of all streamers) at the same time distance of the disconnection (considered as time 0) are considered and the median value is kept. Fig. 3 represents those median values around the disconnection (10 minutes before and 20 minutes after). About three minutes before the disconnection, the spectators start to leave the stream at a high pace. That value is therefore chosen as the start of the OD state. The leaving rate then decreases and the end of the OD state is not that clear from the data: some spectators keep on chatting on a channel hours after it ended. We chose 15 minutes as the end of the OD state. It corresponds to 30% of the maximum number of spectators who are still connected. The now formally defined OD interval is emphasized in Fig. 3.

That result justifies the existence of a new OD state in the transition graph of a streamer. The more complex transition graph supports a finer analysis of the behavior of the spectators: leaving a channel in the OD state is semantically different from leaving when it is either in the ON or the OFF state.

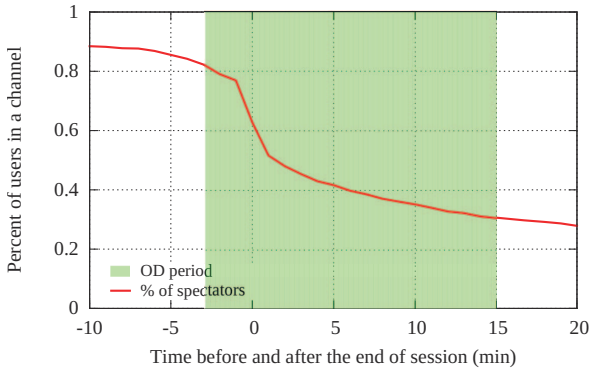


Figure 3: Spectators leaving a channel during the OD state.

D. Understanding the Streamers

The analysis in this section aims to better understand the behavior of the streamers and to see to what extent the content they produce differ from other kinds of online entertainment.

1) Session length: The lengths of the streaming sessions are an aspect of the content broadcast on Twitch.tv. It has already been studied in [1] but we focus here on the popular Starcraft II channels, whereas [1] studied the whole range of video games played by streamers of any level. therefore is worth doing, because it targets a different streamer population.

The average session length in the dataset lasts 5.2 hours. Its median is 3.7 hours (as stated in Table II). In [1] the session median was 1.58 hours. The discrepancy probably reflects the difference of player levels. The professional content, studied in this article, tends to last longer.

Fig. 4 shows the average length duration of each type of channel: individual player or not (teams or TV broadcasting competitions). Such type was manually inferred through a visit to every studied channel. Only 13% of the sessions last less then 2 hours. Most of the sessions last between 2 and 4 hours. The longest sessions (by far: 16 hours on average) come from TV channel broadcasting the World Championship Series[4].

It is interesting to notice how much watching video games differ from other popular kinds of online entertainment. As we have just seen, video game live-streaming produces long duration content in the same online world where YouTube's videos last 4 minutes and 12 seconds on average [15] and Twitter's messages are, at most, 140 characters long.

2) Streamer assiduity: Fig. 5 shows distributions of the average number of sessions per day. The maximal value that

[4]http://wcs.battle.net/sc2/en

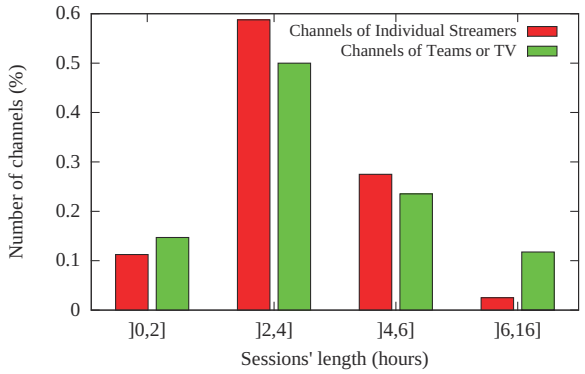


Figure 4: Distributions of sessions' length w.r.t. the type of channel.

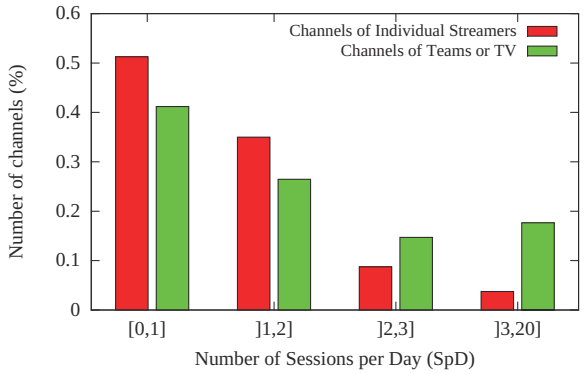


Figure 5: Distribution of sessions per day

is observed is an average of 19 sessions per day. Team and TV channels tend to broadcast more sessions than individual channels. That is expected since they stream more than one player. In fact, the higher the number of sessions per day, the more likely we are dealing with a team or a TV channel. Almost no individual players manage to stream, on average, more than 3 sessions a day.

Comparing that professional gaming production to that of YouTube emphasizes, again, how different those contents are: professional YouTubers such as Mistery Guitar Man [5] or Smosh [6] only release a few videos per week, whereas 30% of the popular Starcraft II players in our dataset broadcast their games in at least two sessions a day. In that regard, gaming can be said easier than producing YouTube content.

E. Understating the spectator

In this section we try to characterize the spectator behavior as we did with the streamer's.

[5]https://www.youtube.com/user/MysteryGuitarMan
[6]https://www.youtube.com/user/smosh

Sojourns	20, 105, 627	
Type	Median	%
$ON \approx ON$	7.5	82.05
$ON \approx OD$	48.5	5.66
$ON \approx OFF$	189.8	2.94
$OD \approx ON$	115.22	0.04
$OD \approx OD$	1.4	0.65
$OD \approx OFF$	40.20	0.05
$OFF \approx ON$	102.91	0.52
$OFF \approx OD$	175.6	0.03
$OFF \approx OFF$	3.18	8.06

Table III: General sojourn percentages by kind.

1) Channel-surfing: We define channel-surfing as a quick sojourn of a spectator in a streamer channel. Such behavior was clearly observed in our dataset. We evaluated more than 20 million sojourns and Table III shows the percentage of each type (defined as a pair entry and exit states). The first column shows the states of the streamer when the spectator joined and left his or her channel. The second column shows the median time of the sojourn. The last column displays the percentage of each kind of sojourn.

The sum of $ON \approx ON$, $OFF \approx OFF$ and $OD \approx OD$ sojourn percentages are equal to 90.8% of the total. Their duration medians are, respectively, 7.5, 3.18 and 1.4 minutes. Despite that, about 20% of the $ON \approx ON$ sojourns and 30% of the $OFF \approx OFF$ sojourns lasted less than a minute. These data indicate that this is a significant behavioral pattern among spectators, and suggest that spectators use Twitch.tv without knowing *a priori* the content they want to consume.

The $OD \approx OD$ sojourn is associated with the shortest duration median (1.4 minutes), which can be explained by the own meaning of the state. When someone joins a channel and the streamer is finishing the transmission, spectators will tend to leave, as they won't be able to see the streamer playing. The $OFF \approx OFF$ sojourn small median also has an obvious interpretation. After joining the streamer channel and finding out that he or she is offline, the spectator quickly leaves the stream.

2) Channel switching: Each spectator seeks to consume a content that pleases him or herself. While watching a given channel, a spectator may decide to switch to another channel in order to find a content that better suits his or her tastes. In this scenario of channel switching, it is important to understand the context where switching happens and what it does mean.

To achieve that, we analyzed our dataset and found exactly 2,386,972 switches during the 139 days during which the data was collected. We summarized the types of switches in Table IV. The first column shows the states of the streamers at origin and destination. The second column shows the median of the spectators sojourn time before making the switch - we used the median because the average sojourn

Kind	Sojourn duration	#	%
$ON \rightarrow ON$	7	1,722,321	72%
$OD \rightarrow ON$	27	292,045	12%
$OFF \rightarrow ON$	20	180,874	8%
$ON \rightarrow OFF$	10	59,781	3%
$OFF \rightarrow OFF$	9	54,864	2%
$ON \rightarrow OD$	6	47,402	2%
$OD \rightarrow OFF$	30	10,156	0%
$OD \rightarrow OD$	15	9,473	0%
$OFF \rightarrow OD$	12	4,817	0%
Total:	—	2,386,972	100%

Table IV: General data about channel switching.

time is distorted by extreme values. The absolute number of channel switches (#) and the corresponding percentage (%) are displayed in the last two columns.

We believe that the state the streamer is when the channel switching happens is related to the reason behind the switch. Understanding this reason is important to characterize how and when streamers attract their spectators, and in order to determine how popular a streamer is - whitch is extremely interesting in the streamer's perspective, since they profit from being popular [5].

For example, a possible interpretation of the $ON \rightarrow X$ switch is that an spectator leaves a session that is not satisfying and goes to another streamer channel. A good analogy to such an action is television: if a spectator is watching a show and in the middle of it changes to another channel, he is probably not enjoying the show very much. Notice that the largest number of channel switching is from the $ON \rightarrow X$ kind, what is explained by the visibility that popular streamers channels achieve in Twitch.tv recommendation system.

To support our hypothesis, in Fig. 6, we plotted a graph comparing, in an $ON \rightarrow ON$ channel switch, how long does a spectator stay in each of the channels. Notice that the spectators tend to stay slightly longer watching the second streamer, supporting the argument that when an $ON \rightarrow X$ happens, the spectator is not satisfied with the content that the first streamer is streaming.

The $OD \rightarrow X$ is the second more popular kind of channel switch - confirming the high spectator evasion described in Section V-C. In this case, assuming that there is a preference for the second streamer does not make any sense from a semantic perspective, once the spectator might be leaving the channel because he or she will not be able to consume its content anymore. This same statement is also valid for switches from $OFF \rightarrow X$ kind, where a spectator, unable to consume the content of one channel, switch to another one.

3) Spectator assiduity: We here define and measure how assiduous the spectators are to the streamers they watch. By "assiduity", we mean a propensity for a spectator e to stick to a small number of streamers among the set S of all the

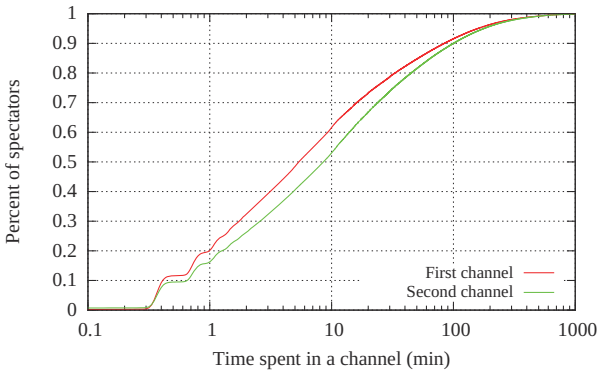


Figure 6: Spectators spend more time in a channel after a switching.

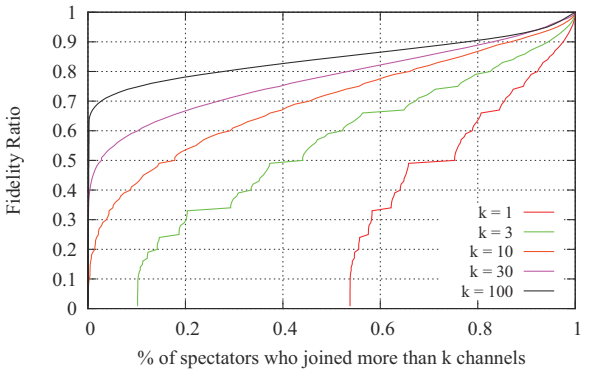


Figure 7: Cumulative distributions of the fidelity ratios of all spectators who joined at least k times.

streams studied in this article. More formally, and noting $incs(e, s)$ the number of times the spectator e joined the channel of a streamer s, the following function F, called *fidelity ratio*, is here proposed:

$$F : e \mapsto 1 - \frac{|\{s \in S \mid incs(e, s) \neq 0\}|}{\sum_{s \in S} incs(e, s)} \quad (1)$$

F maps every spectator e to a number in $[0; 1)$. The higher $F(e)$, the more assiduous the spectator e. The metric is obviously not trustworthy for spectators with few visits to Twitch.tv. In particular, a spectator with one single connection necessarily has a null fidelity ratio.

In Fig. 7, each curve relates to a value k and shows the cumulative distribution of the fidelity ratios of all spectators who joined at least k channels. Those curves confirm what the intuition tells: the more channels joined, the more assiduous the spectator, i.e., the spectators coming again and again on Twitch.com do so to always watch the same streamers. What the figure does not show is that most spectators join very few channels. Among the $1,490,121$ spectators, $455,009$ ($\approx 30\%$) joined more than 3 channels

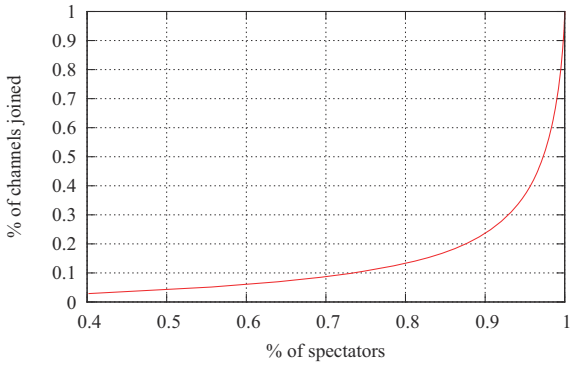


Figure 8: Cumulative distribution of the connections to all channels over the spectators.

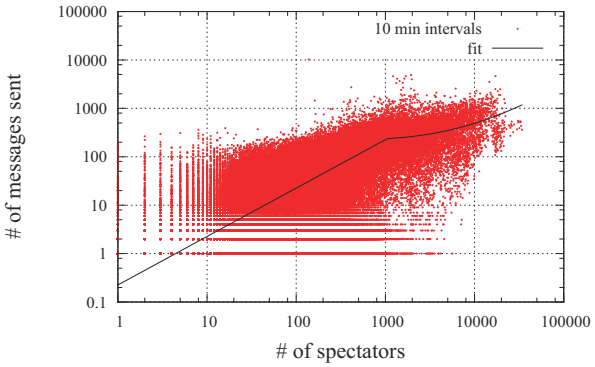


Figure 9: Cumulative distribution of spectators per fidelity ratio.

and only $279,005$ ($\approx 1.87\%$) joined more than 10 channels.

Fig. 8 shows the cumulative distribution of the connections to all channels over the spectators. The top 10% of the spectators are responsible for almost 80% of the connections. More generally, most of the content streamed from Twitch.tv is watched by a small group of passionate spectators. The same small group that was shown earlier to be assiduous (high k value).

4) Chatting: We analyze here the relationship between the number of spectators logged into a Twitch.tv chat and the activity of that chat, i.e., how fast the conversation. More precisely, the activity y is here measured as the number of messages sent in a 10 minutes interval and the related number of spectators x is taken at the end of that time period. We consider all ten minutes intervals in all the channels we collected.

The relationship between the number of spectators (explanatory variable) and the activity (response variable) can be modeled by regression. The curve we propose to fit is composed of two straight lines. The first one starts at $(0;0)$ (because a chat with no spectator cannot have activity) and, at some point, the second line starts where the first one ended. Mathematically, here is the model (where three parameters need to be estimated: a, b and c):

$$y = \begin{cases} a \times x & \text{if } x < c \\ b \times x + (a-b) \times c & \text{if } x \geq c \end{cases} \quad (2)$$

The intuition behind that model is that after certain number c of spectators logged into the chat and a related high chatting activity ac, it becomes harder to communicate and the activity enters a second regime where the growth of the activity w.r.t. the number of spectators is not as intense ($b < a$).

The breaking point c is estimated by an exhaustive search in $[1;1000]$. For each such estimation, a minimization of the squared residuals allows to fit the first line to the points with abscissas smaller than c and the same method is used to fit the second line to the points with abscissas greater or equal to c. The overall sum of the squared residuals conditions the choice the best estimations for c: the smaller the sum, the better. Fig. 9 shows all data points and the model that fits them. The change of regime happens at approximately $c = 1,000$ spectators. At that abscissa, there are $ac = 22.5$ messages that are sent every minute.

To validate the model, its *rms of residuals* is compared to a more naive model that passes by $(0;0)$ and has the same number of parameters: $y = ax^3 + bx^2 + cx$. The model we proposed is better: its *rms of residuals* is 50.63, whereas that of the cubic fit is 53.06.

VI. Conclusion

In this paper we presented a generic model that can be used to analyze video game live-streaming and used it in order to characterize the Starcraft II sub-community (which is mainly a part of the e-sport sub-community). We analyzed data from Twitch.tv, gathered using the website's own API and an IRC crawler. This paper has shown, among other results that: (i) spectators have clear behavioral patterns such as channel surfing and leaving quickly close to the end of a streaming session, (ii) the content produced is longer and less edited than most of online entertainment content, (iii) there is semantic meaning to channel switching (iv) the content is mainly consumed by a small fraction of very assiduous streamers, (v) we can predict the number of messages sent in the chat using a closed formula. Those results are of major interest for all members of the gaming community and also for the scientific one.

As future work, we intend to extend our model to other games that are being broadcasted in Twitch.tv and also check its applicability to other media, such as Youtube. We also intend to design, implement and evaluate novel algorithms that exploit the knowledge of our characterization provides about the behavior of both streamers and spectators, such as recommending streamers in real time.

ACKNOWLEDGMENT

This work was partially supported by CNPq, CAPES, FAPEMIG, and InWEB.

REFERENCES

[1] M. Kaytoue, A. Silva, L. Cerf, W. Meira, Jr., and C. Raïssi, "Watch me playing, i am a professional: A first study on video game live streaming," in *Proceedings of the 21st International Conference Companion on World Wide Web*, ser. WWW '12 Companion. New York, NY, USA: ACM, 2012, pp. 1181–1188. [Online]. Available: http://doi.acm.org/10.1145/2187980.2188259

[2] (2014-05-15) Twitch Named Best Games Related Website for 3rd Consecutive Year in 18th Annual Webby Awards. [Online]. Available: http://www.businesswire.com/news/home/20140429006770/en#.U3UikPgc3Vs

[3] (2014-05-15) How Big Is Twitch's Audience? Huge. [Online]. Available: http://www.forbes.com/sites/davidewalt/2014/01/16/twitch-streaming-video-audience-growth/

[4] T. L. Taylor, *Raising the Stakes : E-Sports and the Professionalization of Computer Gaming*. MIT Press, 2012.

[5] T. Smith, M. Obrist, and P. Wright, "Live-streaming changes the (video) game," in *Proceedings of the 11th European Conference on Interactive TV and Video*, ser. EuroITV '13. New York, NY, USA: ACM, 2013, pp. 131–138. [Online]. Available: http://doi.acm.org/10.1145/2465958.2465971

[6] (2014-2-05) Twitch is 4th in Peak US Internet Traffic. [Online]. Available: http://blog.twitch.tv/2014/02/twitch-community-4th-in-peak-us-internet-traffic/

[7] G. Cheung and J. Huang, "Starcraft from the stands: Understanding the game spectator," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '11. New York, NY, USA: ACM, 2011, pp. 763–772. [Online]. Available: http://doi.acm.org/10.1145/1978942.1979053

[8] K. Sripanidkulchai, B. Maggs, and H. Zhang, "An analysis of live streaming workloads on the internet," in *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '04. New York, NY, USA: ACM, 2004, pp. 41–54. [Online]. Available: http://doi.acm.org/10.1145/1028788.1028795

[9] J. D. Weisz, S. Kiesler, H. Zhang, Y. Ren, R. E. Kraut, and J. A. Konstan, "Watching together: integrating text chat with video," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '07. New York, NY, USA: ACM, 2007, pp. 877–886. [Online]. Available: http://doi.acm.org/10.1145/1240624.1240756

[10] D. A. Shamma, L. Kennedy, and E. F. Churchill, "Watching and talking: media content as social nexus," in *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval*, ser. ICMR '12. New York, NY, USA: ACM, 2012, pp. 12:1–12:8. [Online]. Available: http://doi.acm.org/10.1145/2324796.2324811

[11] P. Cesar and D. Geerts, "Past, present, and future of social tv: A categorization," in *Consumer Communications and Networking Conference (CCNC), 2011 IEEE*, Jan 2011, pp. 347–351.

[12] L. Oehlberg, N. Ducheneaut, J. D. Thornton, R. J. Moore, and E. Nickell, "Social tv: Designing for distributed, sociable television viewing," in *Proc. EuroITV*, vol. 2006, 2006, pp. 25–26.

[13] S. R. Foster, S. Esper, and W. G. Griswold, "From competition to metacognition: designing diverse, sustainable educational games," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2013, pp. 99–108.

[14] J. J. Thompson, M. R. Blair, L. Chen, and A. J. Henrey, "Video game telemetry as a critical tool in the study of complex skill learning," *PLoS ONE*, no. 9, 2013. [Online]. Available: http://www.plosone.org/article/info:doi/10.1371/journal.pone.0075129

[15] (2014-03-10) YouTube statistics. [Online]. Available: http://www.sysomos.com/reports/youtube/

A User Interface Stereotype to build Web Portals

Sofia Larissa da Costa
Instituto de Ciências Matemáticas e de Computação (ICMC)
Universidade de São Paulo (USP)
São Carlos - Brazil
sofialc@icmc.usp.br

Valdemar Vicente Graciano Neto
Instituto de Ciências Matemáticas e de Computação (ICMC)
Universidade de São Paulo (USP)
São Carlos - Brazil
Instituto de Informática (INF)
Universidade Federal de Goiás (UFG)
Goiania - Brazil
valdemarneto@icmc.usp.br

Juliano Lopes de Oliveira
Instituto de Informática (INF)
Universidade Federal de Goiás (UFG)
Goiânia - Brazil
juliano@inf.ufg.br

***Abstract*—Software Engineering for Web Systems domain is a complex process where reuse and productivity are desirable attributes. It involves, among other aspects, modeling user interface (UI) software and its binding to underlying applications business logic and process. Despite recent advances, UI engineering for this domain is still expensive, laborious, and error-prone. On the other hand, Model-Driven Development (MDD) has emerged as a paradigm to bridge reuse and productivity gaps by means of abstract models and automatic software generation through model transformations. However, model-based UI engineering (or MDD for UI) is still an emerging discipline. This paper applies a recent MDD for UI research advance, namely the UI Stereotype, to the UI engineering in the Web Systems domain. The UI Stereotype captures UI specificities, modeling recurrent UI presentation and behavior, abstracting users' interactions and tasks. We apply this concept to describe a Web Portal UI Stereotype as a recurrent interaction pattern that enables the automatic generation of many Web Portal UI components based on model-driven practices. This UI generation approach is compliant with recent advances in UI construction, such as Interaction Flow Modeling Language (IFML), a UI modeling language recently standardised by OMG. Moreover, our approach reduces Web Portals UI software development time-to-market, efforts and costs, contributing to both quality and productivity, and improving maintainability of Web applications.**

***Keywords*-User Interface; Stereotype; Web Portals; Model-Based User Interface; Model-Driven Development.**

I. Introduction

Web Information Systems construction involves organization and control of complex business information [1]. Web applications range from simple information display and conventional Create, Retrieve, Update and Delete (CRUD) forms to complex and highly interactive social networks, search engines and Web portals. Thus, designing and implementing User Interface (UI) software for Web applications can be an expensive and error-prone process [2].

Engineering UI software involves multidisciplinary skills and deals with challenges that start in modelling the mental profile of Web users and continues through designing attractive layouts and dealing with a plethora of languages, frameworks and related UI technologies [3].

On the other hand, Model-Based UI (MBUI) development is one of the mainstream UI Engineering methods [4]. It is defined as the process of creating and refining high level models for the automatic generation of UI. The use of abstract models enhances the user comprehension of how the requirements are transformed into UI software and facilitates the reuse of UI concepts [5], [6], [7]. Thus, MBUI approaches can hide complexities, such as mapping several models, or using different implementation languages, improving the UI construction process by focusing directly on a conceptual representation of the interactive features [8].

In this sense, MBUI approach for automatic building of Information Systems UI presented in [9] takes into account the behavioral and presentation aspects of that software application domain and overcomes important limitations of current methods, namely the specification of applications behavior in response to user interactions, and the integration of UI and the underlying software.

The main concept of this approach, **UI Stereotype**, captures UI similarities of some specific software application, and models UI presentation and behavior recurrent features. It abstracts users' interactions and tasks and defines the way information is presented and manipulated for each particular task in the UI [9]. In this way, UI Stereotype contributes to reuse and standardization since to model an UI using a known UI Stereotype consists in mapping the Domain Concepts to the UI Stereotype elements.

This paper defines a UI Stereotype for a recurrent way of interaction in Web Systems domain: Web Portals, a gateway to information and services on the Web [10]. Modeling and building Web Portals based on a UI Stereotype is important because of the continuously evolving of Web Systems applications, which must be frequently updated to aggregate information and make it available for several user profiles. The use of the Web Portal UI Stereotype is illustrated with real examples to demonstrate its capability

978-1-4799-6953-1/14 $31.00 © 2014 IEEE
DOI 10.1109/LAWeb.2014.8

of improving Web Portals building with the application of the MBUI approach of [9].

The remainder of this paper is organized as follows. Section II presents the main concepts and summarizes the proposed approach to build UI. Section III describes the Web Portal UI Stereotype and shows some UI synthesis from it. Section IV discusses the applied approach to develop the Web Portal Stereotype and its correlation to IFML OMG standard. Finally, Section V concludes the paper and points to future works.

II. Model-based Approach to build User Interfaces

Model-Driven Development (MDD) is a software development life cycle [11] where models are the first-class citizens. It addresses system complexity by the intense use of models [12]. Under this paradigm, software development is treated as a set of model transformations that are conducted from the earliest steps of the software development process to the latest stages. After subsequent transformations, the process delivers a deployable, complete and tested software product [13], [14]. It promotes knowledge reuse through models, reduces time-to-market, and increases productivity by means of software automatic generations from model transformations.

In MDD, models must conform to their respective metamodels [15] and transformations are mappings between a source metamodel and a target metamodel [11]. A lot of acronyms have emerged in literature to describe model-driven practices along the last years. Prominent among the MDD initiatives is OMG's Model-Driven Architecture (MDA) [16]. MDA standard defines different model categories as follows [13]:

- Computation Independent Model (CIM), representing the problem domain;
- Platform Independent Model (PIM), representing the solution domain without platform specific details;
- Platform Specific Model (PSM), representing the solution domain with platform specific details.

MBUI development is a MDD approach that uses metamodels to specify essential characteristics for the interaction design through a high-level UI description [17], [18]. The CAMELEON reference framework establishes the basis for this discipline and it is a MDA-compliant approach [3], since this framework classifies the involved models in a similar way. This framework decomposes UI development in four steps: (1) Modeling Domain Tasks and Concepts, considered as CIM; (2) Definition of Abstract UI containers and components, interpreted as PIM; (3) Concretization of UI components, considered as PSM; (4) and Final UI generation, that corresponds to the executable code [4], [3].

The MBUI approach for automatic building of UI applied in this paper [9] uses this framework, disconsidering task modeling, which is a well established technique [19]. Three metamodels are used to describe ISUI [9]:

1) Domain Metamodel, that corresponds to domain concepts;
2) HCI Metamodel, used to define the abstract UI;
3) Presentation Metamodel, that represents the UI in a concrete level.

The first step in this approach, *Modeling Domain Concepts*, is done by building an instance of the Metamodel Domain. It is based on the UML Class Diagram and defines the main Web Information System (WIS) domain concepts and it is used to model domain concepts of business data and associated rules to be applied by WIS, as reported in [20].

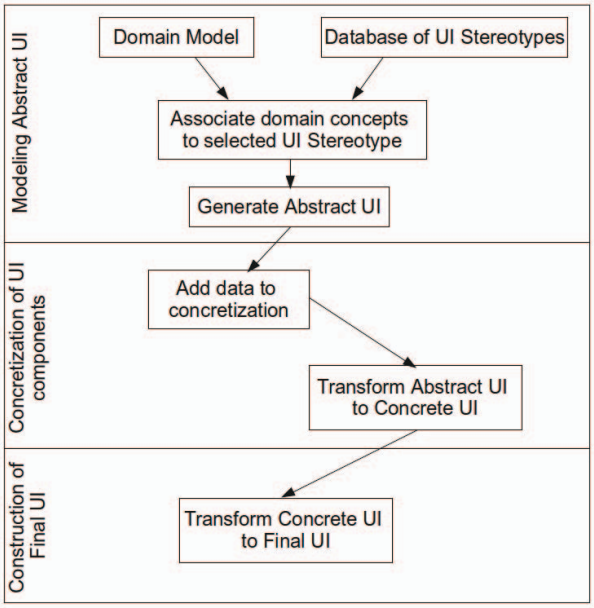


Figure 1. Process to model-based generation of UI

The other steps of the reference framework – defining *Abstract UI*, *Concrete UI* and *Final UI* – involve UI modeling and construction. Figure 1 summarizes this steps of the approach to build a WIS User Interaction (UI). For this, we use the UI Stereotype concept [20], [9], that should be a dominant element in UI which is an abstraction of organization and type of UI elements as well as the proper behavior of UI which perform the same tasks and present the same type of information and interaction.

The second step is modeling abstract UI, as seen in Figure 1. In this sense, the HCI Metamodel aims to describe presentation and behavioral characteristics of WIS UI in abstract level. These UI make intensive use of information stored in databases and organize tasks related to business processes. The key concept of HCI Metamodel is the User

Interface Stereotype, or simply **UI Stereotype**, which consists of an abstraction of the UI intention, independent of the application or underlying software. The UI Stereotype enables to abstract interactions and tasks of the user as well as the form of presentation of information for a particular task. Thus, each stereotype is a UI dominant element, which organizes the presentation of the subordinate elements and its behaviors [9]. An instance of UI Stereotype contains at least one UI Element, according to the taxonomy of UI Elements [9]. Also, the UI actions are predefined by the UI Stereotype. Thus, all instances of a UI Stereotype execute the same behavior. Actions which depend on the application or other external system are specified as external actions and are associated to an instance of UI Stereotype in this step.

In this way, to obtain an abstract UI we select an existent UI Stereotype to associate concepts of business element in the Model Domain which should be in the UI. If the given task do not have a UI Stereotype, we should model it in this step. Using a UI Stereotype, software engineers only configure each HCI element and the actions according to the selected stereotype. In other words, since we select an existent UI Stereotype, we should associate each domain concept to an UI element in the UI Stereotype. Also, the external actions are associated to the correspondent application.

The third step is to obtain the concrete UI, through Presentation Metamodel, which enables UI description in a concrete level, i.e. the target computational platform. It is worth noting that the Presentation Package of the HCI Metamodel deals with abstract definitions of UI appearance, while the Presentation Metamodel aims on mapping both abstract appearance and behavior to concrete UI components to allow code generation [9]. In this way, the obtained model in this step contains data regards to the target platform. Similar to the previous step, if a UI Stereotype is selected, there must be a template in the target computational platform that represents the UI Stereotype selected in a concrete level.

Finally, the fourth step is to generate the final UI in runtime, which is done by a tool that generates the final UI code using the obtained models. However, it is important to obtain a set of UI Stereotypes in order to use this approach. [20] presents the CRUD Stereotype and [9] presents the Survey Stereotype. In this sense, this paper presents another UI Stereotype: Web Portal. We focus on the first three steps. Next section details building of Web Portal Stereotype.

III. Web Portal UI Stereotype

A Web Portal is a special Internet site designed to act mainly as a gateway to give access to other sites [10]. It offers centered access to relevant content and applications for a given interest, allowing the users to access and interact with business elements, applications and processes. Also named as Enterprise Information Portal, a Web Portal can provide access to a wide variety of information about an organization [21], [22], helping users to find products, services and information from this organization in the Web.

The first step to define a UI Stereotype as Web Portal is to model the domain concepts. Figure 2 summarizes the main domain concepts used in a Web Portal. The main business element in this domain is the Web Portal, which brings together information, services and applications regarding a specific organization. Some data of the organization is required, such as name, address, contacts and logo (that should be a graphical element). A Web Portal presents subsites and services to users. There is a rule that defines how to execute a service.

The second step to generate a UI Stereotype is to obtain the HCI Model, that presents the UI Elements required and the organization of them in the proposed UI Stereotype. Figure 3 presents an abstract view of Web Portal Stereotype, adapted from [22]. In this view, a Web Portal is divided into:

1) Top: identification of organization;
2) Horizontal Navigation: it allows to access subsites and subsystems;
3) Vertical Navigation: it allows to execute specific tasks in the subsystems.
4) Content: presents the content of the system or adds other stereotypes, such as Survey [9];
5) Footer: informative data about the developed system.

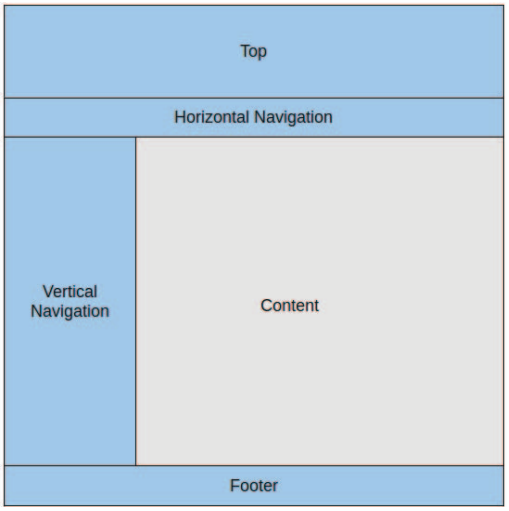


Figure 3. Abstract View of Web Portal Stereotype

In addition, we note that in recent years the initial page of Web Portals has a different view, ommiting the vertical navigation and inserting in central part the main tasks and tasks of the system. Figure 4 shows the abstract view of initial page.

In this way, we modelled the Web Portal as an instance of HCI Metamodel. Figure 5 presents the HCI elements for a Web Portal Stereotype. Moreover, this model presents the

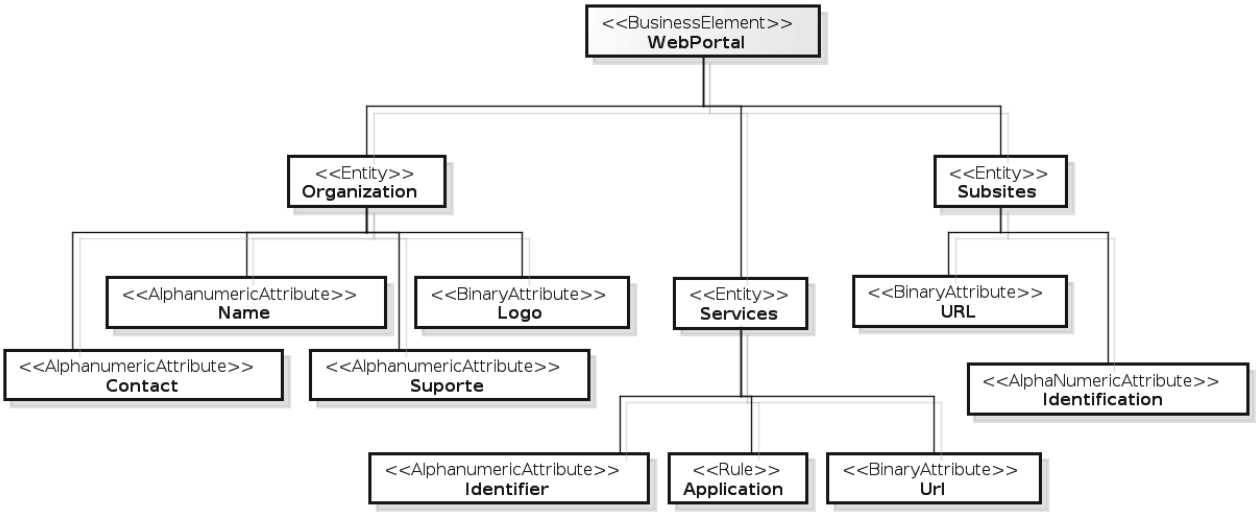


Figure 2. Domain Model of Web Portal

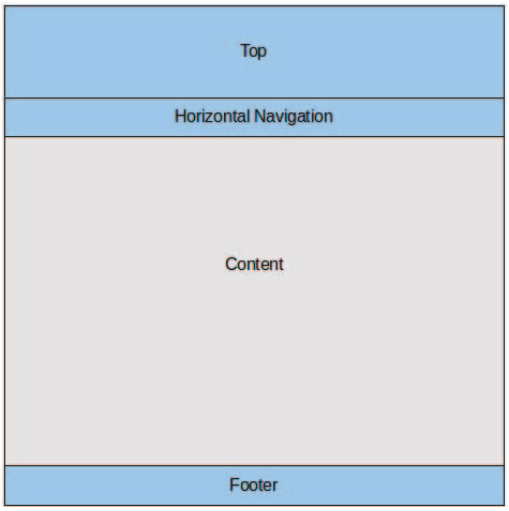


Figure 4. Abstract View of Initial Page of Web Portal Stereotype

type of action that should be executed in this stereotype, i.e. Interface Actions since these actions allows the navigation between pages of each system.

The next step is generating the Presentation Model of Web Portal. This step consists of a mapping between the HCI model and Presentation Model. Figure 6 presents the Presentation Model for Web Portal. This figure represents the concrete classes that will be generated to execute a Web Portal in a web browser. Layout, that is related to UI panels position, is described in a JSF file. Moreover, each panel points to a CSS (*Cascading Style Sheets*) class, that describes the appearance format. The class WebPortal should have a method that allows the navigation between pages, through the Interface Rule executed in menu items.

A. Example

Figure 7 presents an example of Web Portal for Instituto de Informática of Universidade de Goiás[1], in Brazil. Figures are in Portuguese, but we analyze the structure and organization of them. In Figure 7 we note that the structure and organization is similar to the abstract view shown in Figure 3. The identification of the institute is on top of the INF Web Portal. Follow the top, we note a horizontal navigation with some subsites. On the left of page, there is a vertical navigation bar categorized according to interest information.

However, this portal is a subsystem of the Universidade de Goiás[2], and its initial page is seen in Figure 8. This page is similar to abstract view of a initial page of a Web Portal, as seen in Figure 4, since no exists the vertical navigation on left.

Figure 9 shows the HCI model for INF Web Portal. This figure summarizes the abstract HCI elements of the INF Web Portal. This elements can be mapped to the Presentation Model and them to the application that is presented in Figure 7.

IV. Discussion

Web Portals are ubiquitous Web applications. This paper shows that the UI of this kind of application has the same intention and can be modelled as a single UI Stereotype, promoting reuse of presentation and behavioral aspects. [9] shows how UI Stereotype is distinct from other MBUI development proposals for Web Systems generation.

[1]http://www.inf.ufg.br/
[2]http://www.ufg.br/

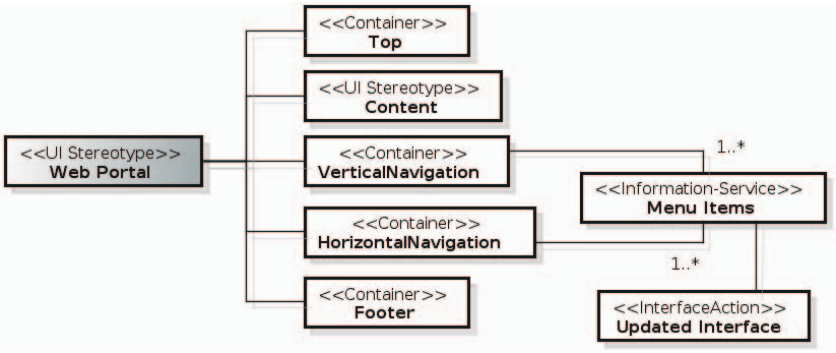


Figure 5. HCI Model for Web Portal Stereotype

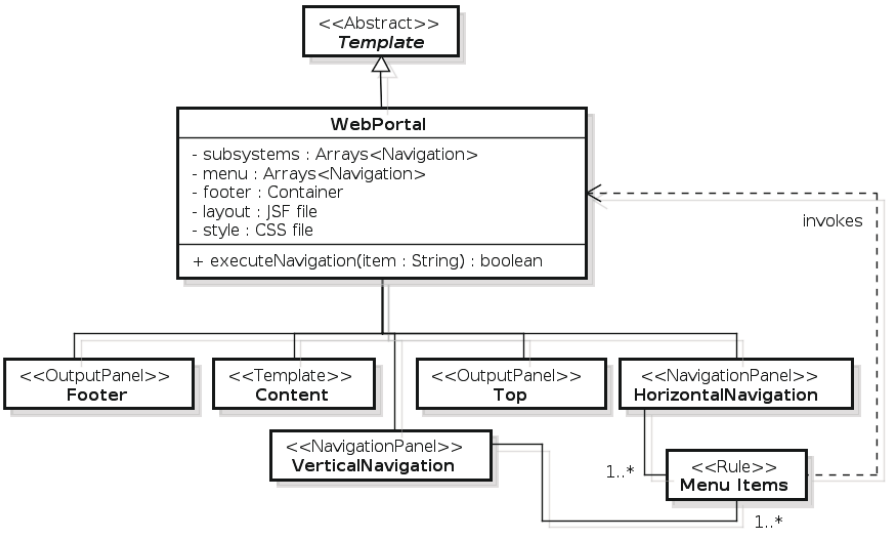


Figure 6. Presentation Model for Web Portal Stereotype

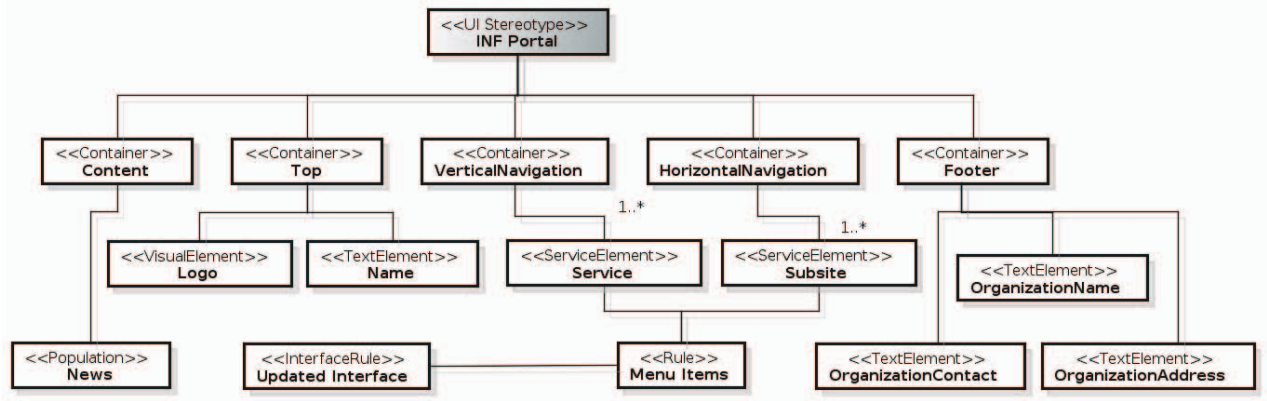


Figure 9. HCI Model for INF Web Portal

Meanwhile, Interaction Flow Modeling Language (IFML) has emerged as an OMG standard [3] and consists of a model-

[3]http://www.omg.org/

Figure 7. Example of Web Portal

Figure 8. Example of Initial Page of Web Portal

driven approach to express the content, user interaction and control behavior of the front-end in (Web) software applications [23]. Therefore, IFML has concerns that also appear in our approach, but the UI Stereotype is not a substitute for IFML. Table I presents a comparison between IFML and the MBUI Stereotype approach we propose in [9]. In fact, we could adapt our metamodels to cope with IFML since the UI Stereotype is an abstraction that can be constructed over an IFML model, taking advantage of tools supporting this language, such as WebRatio [4].

IFML is divided in three packages: the Core package, the Extension package and the DataTypes package. The Core package contains the concepts that build up the interaction infrastructure of the language in terms of InteractionFlowElements, InteractionFlows and Parameters. Core package concepts are extended by concrete concepts in the Extension package with more complex behaviors. The DataTypes package contains the custom data types defined by IFML [23].

UI Stereotype is an abstraction that captures UI similarities in presentation and behavior aspects to model recurrent front-end of software applications. Thus, UI Stereotype is closer to a *pattern for HCI*, since it documents a prescription for a recurrent problem (UI modeling), a context and a systematic repeatable solution. IFML focuses in the structure and behavior of the application as perceived by the end user [23]. The MBUI approach using UI Stereotype can be seen as a concrete initiative in conformance to IFML principles which offers a concrete collection of materialized recurrent visual interaction patterns. A UI Stereotype, such as Web Portal, offers a complete set of presentation appearance and expected behavior, as recommended by the IFML specification.

IFML specification classifies itself as a PIM [23]. The MBUI approach using UI Stereotype could be seen as a PSM for IFML that meets the classic features of a standard user interaction: the visual structure, the navigation model,

[4]http://www.webratio.com/

Table I
COMPLIANCE ANALYSIS

Quality Criteria	MBUI approach to build ISUI (PSM) [9]	IFML [23] (PIM)
Beautification	It supports via View Container.	CSS Insertion in Presentation Model with linking to a concrete View Container element.
Layout	Modeled as a diagram of View Containers.	Template in JSF
Separability	Supported by different Action Concepts	Use of different models
Intention	Implemented in concrete level	A set of UI Stereotypes
Decomposition		Taxonomy of UI elements in HCI Metamodel
Standardization	Supported by a grouping of View Containers	UI Stereotypes
Clarity	Supports once UI Stereotype is in accordance with IFML	Mapping rules to different models
Flexibility	MDD transformation technologies enables that and IFML public metamodel allows it to be done.	Information addition in concrete level
Direction	Mapping between IFML and our model encompasses that	Mapping rules
Generality	Totally supported by IFML once it abstracts UI structure, navigation, and business association	HCI Metamodel
Structure	UI concepts are documented in IFML metamodel	Metamodels described in UML
Contextualization	Implemented in concrete level	Mapping Domain Metamodel to HCI Metamodel
Correlability	Totally supported	Integration of the three models

and the underlying software behavior through business rules and actions. To elucidate similarities, an excerpt of IFML metamodel is described in Figure 10 (the View Elements). It presents the View Containers that we use to argue, in Table I, that we can accomplish our respective concrete implementation conforming to the IFML metamodel.

Table I shows an adaptation of a comparison made between the UI Stereotype concept and other user interaction representation initiatives presented in [24]. This table takes into account a set of quality criteria organized in a taxonomy in accordance with presented challenges for MBUI approaches in [24]. Thus, a MBUI approach should satisfy these criteria to be considered a suitable MBUI approach. Considering IFML as a PIM and our approach under a PSM perspective, Table I presents how IFML supports those features which a MBUI approach should satisfy, and how the UI Stereotype approach implements the respective features in an implementation.

The requirements were divided into a taxonomy of five classes related to MBUI approaches (such as IFML) established in [24]: Organization, Decomposition, Mapping, Abstraction and Compliance.

Organization brings the requirements associated with the organization of UI building, and its features are Beautification, Layout and Separability. Beautification regards to embellished and refinement of UI appearance support throughout its life cycle. It involves details such as the type and font size, background color, icon images, placement of components, among other aesthetic details. Layout specifies elements grouped and organized in the layout and its arrangement. Separability regards to separation of concerns in the approach, although application domain concepts involved. This separation promotes maintenance of business concepts and UI in isolation.

Requirements related to Decomposition classify the subdivision of UI components: Intention, Decomposition and Standardization. There are UI for different purposes, and the Intention feature should bring together a set of UI application intentions that can assist in building productivity UI. One recurrent user interaction intention being used in WIS is the Web Portal. The concept of dominant element, the one that makes the grouping of simple elements to construct more complex elements, can solve recurrent problems in UI, and is abstracted in Decomposition feature.Isolated elements do not have a purpose for use. However, together they become reusable in similar contexts, fostering productivity in UI building and promoting the Standardization of the appearance and behavior of the UI.

Mapping requirements are related to the need of transformation of models for automated UI construction. Clarity, Flexibility and Direction are features of Mapping. Clarity relates to the efficient use of transformation rules between the abstract and concrete metamodel that must be clear and precisely specified. In addition, some components can be more abstracts than one mapping for concrete elements, and Flexibility must be allow to configure this mapping. The Direction is observed when the mappings are done in a proper way, from a more abstract to a more concrete representation.

Abstraction presents features that support an independent technology approach. Generality is related to maintaining a high level abstraction and generality in the models used for specifying UI, making independent technology specification. Structure is related to providing a set of precisely defined terms on IHC, covering concepts of presentation and behavior of UI.

The Compliance requirements are related to building UI according to the application need. For this, Contextualization and Correlability characteristics are expected. Correlability involves the association between features and business con-

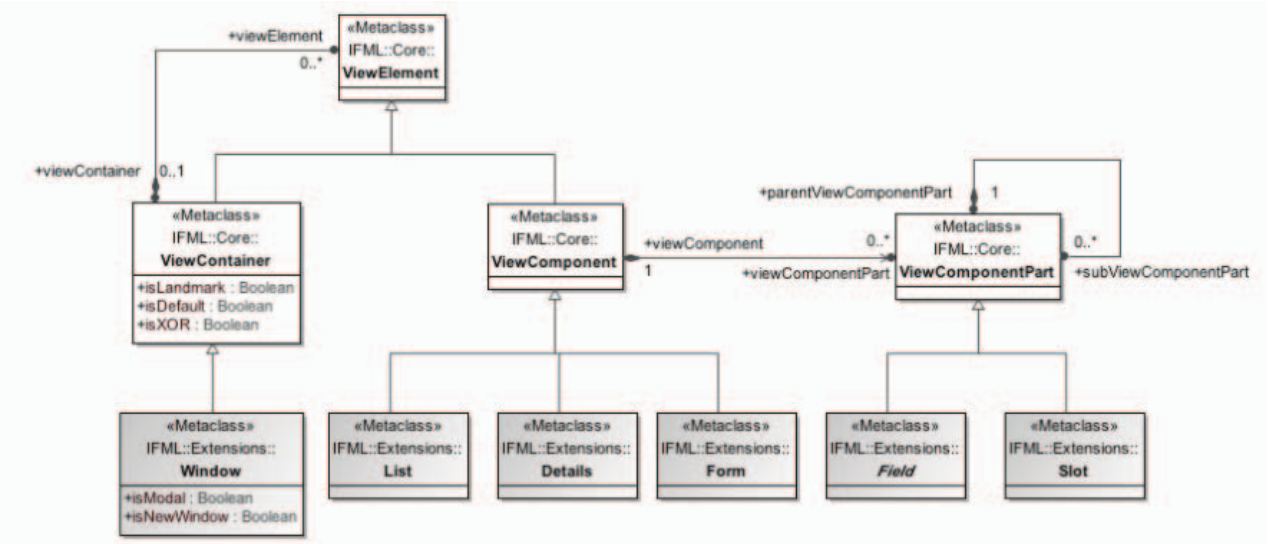


Figure 10. IFML View Elements - Excerpt of IFML Metamodel.

cepts with the interface specification, allowing to understand which business concepts and features must be present in the UI [25]. The Contextualization is concerned with the appropriate presentation of the UI according to the business domain information types and relationships.

Thus, our approach complies with IFML standard and is suitable to address Web Systems modeling and automatic UI generation issues. Furthermore, in a comparative analysis, existing alternative approaches do not support all requirements presented in Table I [9]. Other recent approaches, such as CIAT [26] and Malai [27], do not use a dominant element that standardizes UI which have the same type of information and the same purpose of interaction. In spite of the advantages of use UI Stereotype, we do not control how UI Stereotype is modeled. Moreover, modeling a UI Stereotype which is rarely used can be onerous to the UI development process.

V. Conclusions

This paper describes a UI Stereotype to improve the construction of Web Portals using a model-based approach to build the UI for Web Systems domain. The proposed Web Portal UI Stereotype can generate different Web portals with the same intention and from a common set of metamodels, fostering reuse in User Interaction (UI) Engineering. This approach allows programmers only configure each HCI element in a UI Stereotype instance. The stereotyped behavior should be aggregated to the tool (as WebRatio) so programmers should need to implement only specific functions of an IS. Thus, the use of a UI Stereotype that describe Web Portals appearance and behavior promotes the standardization of this kind of application.

As future directions, we will migrate the concept of UI Stereotype for IFML standard, taking advantage of the underlying tool support, fostering reuse through the abstraction of a UI Stereotype over the IFML model. Also, we will apply the proposed UI Stereotype in the development process of Web Portals to observe how to improve this approach and how programmers react to the use of this method.

We also intend to investigate the description of UI Stereotypes as Design Patterns [28], [29] for UI Engineering, since the collection of UI Stereotypes a(such as Web Portal, Survey, and CRUD) can be seen as recurrent interaction UI challenges for Web applications. The solution for these challenges can be structured and documented as a tuple composed by a Recurrent Problem (UI Modelling and Construction), a Context (Web application UI development), and a Repeatable Solution (automatic UI synthesis via Model-Driven Development based on UI Stereotypes).

References

[1] J. L. de Oliveira, L. Loja, S. Costa, and V. Graciano Neto, “A component for business processes management in information systems (in portuguese),” in *SBSI 2011*, Salvador, Brazil, May 2011.

[2] C.-M. Lee, “User Interface Prototype Generation Technique Supporting Usage-Centered Design,” *Intern. Journ. of Software Engg and Knowledge Engg*, vol. 19, no. 1, pp. 23 – 46, 2009, world Scientific Publish. Co.

[3] J. Vanderdonckt, “A MDA-compliant environment for developing user interfaces of information systems,” in *CAiSE 2005*, Porto, Portugal, June 13-17 2005, pp. 16 – 31, springer.

[4] G. Calvary, J. Coutaz, D. Thevenin, Q. Limbourg, L. Bouillon, and J. Vanderdonckt, "A unifying reference framework for multi-target user interfaces," *Interacting with Computers*, vol. 15, no. 3, pp. 289 – 308, 2003, computer-Aided Design of User Interface.

[5] S. Ahmed and G. Ashraf, "Model-Based User Interface Engineering with Design Patterns," *Journal of Systems and Software*, vol. 80, pp. 1408 – 1422, August 2007, elsevier Science Inc.

[6] T. Memmel and H. Reiterer, "Model-Based and Prototyping-Driven User Interface Specification to Support Collaboration and Creativity," *Intern. Journal of Universal Computer Science - Special Issue on New Trends in HCI*, vol. 14, no. 19, pp. 3217–3235, March 2009.

[7] J. Falb, R. Popp, T. Rock, H. Jelinek, E. Arnautovic, and H. Kaindl, "Fully-automatic generation of user interfaces for multiple devices from a high-level model based on communicative acts," in *Hawaii ICSS 2007*. Washington, DC, USA: IEEE Comp. Society, 2007, pp. 26 – 35.

[8] T. J. Bittar, R. P. d. M. Fortes, L. L. Lobato, and W. M. Watanabe, "Web Communication and Interaction Modeling using Model-Driven Development," in *DC 2009*, Bloomington, Indiana, USA, October 5-7 2009, pp. 193 – 198.

[9] S. L. da Costa, V. V. Graciano Neto, B. dos Reis Calçado, and J. L. de Oliveira, "User interface stereotypes: A model-based approach for information systems user interfaces," in *SBSI 2014 (to appear)*, Londrina-PR, May 2014.

[10] A. Tatnall, *Web Portals: The New Gateways to Internet Information and Services*, H. Arthur Tatnall, Ed. Idea Group Publishing, March 22 2005.

[11] V. V. Graciano Neto and J. L. de Oliveira, "Evolution of an application framework architecture for information systems with model-driven development (in portuguese)," in *SBSI 2013*, 2013, pp. 1–12.

[12] W. A. D. Santos, B. runo B. . F. . Leonor, and S. tephan Stephany, "A Knowledge-Based and Model-Driven Requirements Engineering Approach to Conceptual Satellite design," in *ER 2009, Proc.*, vol. 5829, 2009, pp. 487–500.

[13] M. B. Nakićenović, "An agile driven architecture modernization to a model-driven development solution - an industrial experience report," *Intern. Journ. on Advances in Softw.*, vol. 5, no. 3, pp. 308 – 322, 2012.

[14] A. van Deursen, E. Visser, and J. Warmer, "Model-driven software evolution: A research agenda," in *MoDSE 2007*, D. Tamzalit, Ed., Amsterdam, The Netherlands, March 2007, pp. 41–49.

[15] J. Canovas and J. Molina, "An architecture-driven modernization tool for calculating metrics," *IEEE Software*, vol. 27, no. 4, pp. 37–43, 2010.

[16] "Object Management Group (OMG), Model-driven Architecture (MDA)," http://www.omg.org/mda/, 2003.

[17] J. Vanderdonckt and A. R. Puerta, "Introduction to Computer-Aided Design of User Interfaces," in *CADUI 1999*, Louvain-la-Neuve, Belgium, Oct 21-23 1999, pp. 1–6.

[18] A. Puerta and J. Eisenstein, "Towards a General Computational Framework for Model-Based Interface Development Systems," in *Proc. of the Intern. Conference on Intelligent User Interfaces*. ACM Press, 1999, pp. 171 – 178.

[19] Q. Limbourg and J. Vanderdonckt, *The Handbook of Task Analysis for Human-Computer Interaction*. Mahwah: CRC Press, 2003, ch. 6, lawrence Erlbaum Ass.

[20] S. L. da Costa, V. V. Graciano Neto, L. F. B. Loja, and J. L. de Oliveira, "A Metamodel for Automatic Generation of Enterprise Information Systems," in *BW-MDD 2010*, vol. 8. Salvador, Brazil: UFBA, Setembro 2010, pp. 45 – 52.

[21] M. White, "Enterprise information portals," *The Electronic Library*, vol. 18, no. 5, pp. 354–362, 2000.

[22] M. van Welie, "Interaction Design Pattern Library," available in: http://www.welie.com/patterns/index.php. Last access: February 1 2011.

[23] OMG, *Interaction Flow Modeling Language (IFML)*, Object Management Group (OMG) Std. ptc/2013-03-08, March 2013. [Online]. Available: http://www.omg.org/spec/IFML/1.0

[24] S. L. da Costa, "A model-based approach for automatic construction of user interfaces for information systems (in portuguese)," Master's Thesis, UFG, 2011.

[25] J. Vanderdonckt, "Model-Driven Engineering of User Interfaces: Promises, Successes, Failures, and Challenges," in *ROCHI 2008*. Bucarest, Romania: Matrix-ROM, September 2008, pp. 1–10.

[26] A. I. Molina, W. J. Giraldo, J. Gallardo, M. A. Redondo, M. Ortega, and G. GarcÃa, "Ciat-gui: A MDE-compliant environment for developing graphical user interfaces of information systems," *Advances in Engineering Software*, vol. 52, no. 0, pp. 10 – 29, 2012.

[27] A. Blouin and O. Beaudoux, "Improving modularity and usability of interactive systems with Malai," in *SEICS 2010*. Berlin, Germany: ACM, June 19-23 2010, pp. 115 – 124.

[28] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns*. Boston, MA, USA: Addison-Wesley Longman Pub. Co., Inc., 1995.

[29] T. Neil, "Rich Internet Application Screen Design," *UX Magazine*, no. 483, February 11 2010.

It is not just a picture: Revealing some user practices in Instagram

Camila Souza Araújo, Luiz Paulo Damilton Corrêa
Ana Paula Couto da Silva, Raquel Oliveira Prates, Wagner Meira Jr.
Computer Science Departament
Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte, Minas Gerais, Brazil
Email: {camilaaraujo, luiz.correa, ana.coutosilva, rprates, meira}@dcc.ufmg.br

***Abstract*—In this work we investigate the user practices in Instagram, a social photo sharing service. Some interesting conclusions emerge from our analysis. For instance, users tend to concentrate their posts during the weekend and at the end of the day. Furthermore, people tend to endorse photos with many *likes* and *comments*, inducing the *rich get richer* phenomenon. Our findings can support future research on sociology and cultural analitycs research areas, such as on the proposal of new clustering algorithms based on the user practices in different social media networks.**

***Keywords*-Instagram, Post Popularity, Photography.**

I. Introduction

It is well-known that technology has been changing the way people see the world as well as interact with each other. Among the several technology innovations introduced in the latest years, the smartphone can be considered as one of the most important and impressive ones. The everyday use of smartphones with high quality built-in cameras combined with the online social networks, such as Facebook[1], Youtube[2], Twitter[3], Flickr[4] and Instagram[5], have lead to a new way of sharing and reacting to life events. This new environment brings several interesting questions about people's social behavior and the impact of these new technologies in our everyday life. Some works that explore these questions can be found in [1], [2], [3].

In this work, we focus on user practices in which a photograph application on smartphones is used to share life events experiences. We examine the use of Instagram, a social photo sharing service. Instagram includes dedicated mobile applications that allow users to take and manipulate photographs by adding filters and frames, and to share them online where other users can react through *comments* and *likes*. Launched in October 2010, Instagram has seen enormous growth. According to usage statistics[6], the service has 200 million registered users who have posted so far 20 billion photographs, with an average of 60 million photographs per day. Based on these numbers, Instagram can be considered one of the most popular applications for sharing photos and for interacting with friends, acquaintances and worldwide brands.

The examination of Instagram practices in our work is based on analysis of $1,265,080$ publicly accessible photos and videos posted by ordinary and popular Instagram users, selected either from specialized blogs[7] or randomly selecting an ID from the set of users. We analyze how users react to new posts through like and comment activities. Some interesting conclusions emerge from our analysis. Users tend to concentrate their posts during the weekend and at the end of the day. Moreover, people tend to endorse photos with many likes and comments, inducing the rich get richer phenomenon also presented on online and real-life social networks. From our results it is possible to infer that mobile technologies and social media applications play an important role on changing the relationship between people and photography. Nowadays, photography can be considered a powerful tool for expressing feelings and for telling about important life events to a large number of people.

We believe that the results we present in this paper provide a better understanding of how people interact with photographs in the social media era. Moreover, we think that our analysis reveal important cultural aspects that are beyond the photograph scope, providing valuable material to sociology and cultural analytics[8] research areas.

II. Related Work

Authors in [4] present an initial exploration of several user practices that have evolved around online sharing in websites, more specifically Flickr. They focus on how Flickr practices contrast with more traditional digital photo sharing, named by Richard Chalfen as Kodak Culture [5]. For instance, unlike Flickr users, Kodak Culture people communicate primarily within their existing social groups of friends

[1]http://www.facebook.com
[2]http://www.youtube.com
[3]http://www.twitter.com
[4]http://www.flickr.com
[5]http://www.instagram.com
[6]http://www.instagram.com/press (last access May 2014)
[7]http://web.stagram.com/hot
[8]http://lab.softwarestudies.com/p/cultural-analytics.html

978-1-4799-6953-1/14 $31.00 © 2014 IEEE
DOI 10.1109/LAWeb.2014.12

and family, sharing images of traditional subjects such as birthdays and family holidays. Furthermore, Kodak Culture people want to control the level of storytelling around and the privacy of different photos. Flickr users, instead, use it as a way to document their lives and view photosharing as a fundamentally public act, organizing themselves in different communities around different photographic styles or subjects.

Instagram is the most popular application that combines smartphones with cameras and the possibility of constant access to social media, enabling easy sharing of images of people's lives. Some academics studies are worried about characterizing the application itself without focusing on how users and their friends organize themselves around photos [6], [7]. In our work, instead, the main role is played by the photos themselves. We focus on the photos in order to reveal some cultural practices that take place through Instagram.

Few studies in the literature focus on understanding user practices and culture through the photos posted in Instagram. Hochman and Schwartz [1] analyzed a dataset of over half a million photos taken in New York and Tokyo and used visualization techniques in an attempt to highlight cultural differences. Authors in [2], instead, examined how users manipulate photographs by adding filters and frames, as well as the process of sharing them online where other users can react through comments and likes in a very specific environment: museums. Furthermore, authors rely on a small number of instagram posts ($\approx$ 225), drawing their conclusions by the data provided from interviews with 16 individuals. Here, our analysis is not limited to a specific environment and we are not focused on how users categorize their posts.

III. Dataset

The material used in our analysis englobes a dataset that was collected during February and March 2014. The dataset consists of $1,265,080$ photographs or videos from $256,398$ users. Data was collected using the Instagram Application Programming Interface (API)[9].

As we are interested in a broad type of user profiles, our dataset was collected in two different ways. First, we collected 200 profiles that belong to celebrities, professional photographers and worldwide brands, using specialized blogs[10]. Second, we randomly selected accounts from the users ID application set. In this way, it is guaranteed that our dataset is composed by popular and ordinary Instagram users.

We also collected a set of user features (*Total number of posts, Website, Total number and list of who the user follow, Total number and list of who follows the user - 'followed by' users*) as well as post features (*Identification number, Total number of tags - terms with #, List of tags, Total number of likes, List of users who commented and liked the post, Time and Day, Filter*). These features will enable us to identify user practices and, consequently, possible new trends on how people interact and share photography in the new era of social media networks.

Finally, in our dataset, 48% of the users have neither shared an image nor a video in their accounts. Active users (52%) can be divided into two groups: 85% of them have shared up to 30 posts and 15% of them more than 30 posts. As expected, engaging in Instagram has different levels of intensity. It is very likely that inactive users are those who signed up to the application either for curiosity (when the application was launched) or just for following other people (for instance, celebrities or brands). In the other hand, users that are very active may represent celebrities, professional photographers or brands that may use Instagram as a self-promotion media.

IV. Results and Discussion

We analyzed the data collected regarding how users feed the application and interact with the posts. In this section, we discuss our finding about *users' practices* in Instagram.

Figure 1 shows that Instagram users massively share their photos and videos over two days: Saturday and Sunday ($\approx$ 50% of the posts). We may suppose that users make extensive use of their smartphones to share interesting life events, that are more likely to happen at these particular days. Interestingly, a non-negligible number of posts is shared on Friday and Monday. Maybe a large number of them are related to weekend events.

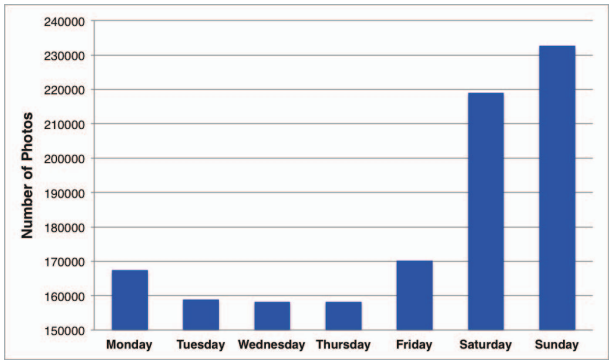


Figure 1. Shared posts over the days of the week.

Focusing on the daily basis shared posts, we note that users have some schedule preferences. Figure 2 depicts at what times users post more frequently. Posts are mainly shared during the afternoon and the evening, with the highest peaks at 21:00. A non-negligible number of posts is also shared overnight.

Next, we look at how users can enrich the image quality of their posts. Figure 3 shows the filters applied by the

[9]http://instagram.com/developer/
[10]http://web.stagram.com/hot

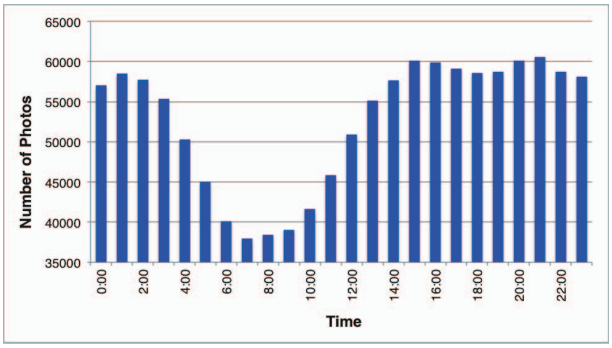


Figure 2. Daily basis shared posts. The Number of Photos axis is ranged in the minimum and the maximum number in our dataset.

users we monitored. In our dataset, the majority of the posts (76%) has some image processing. However, a non-negligible number ($\approx$ 24%) of shared posts is unfiltered (Normal filter or none). The users that shared posts without filters may be amateurs who still cannot work their smartphones. Moreover, photos may be treated outside Instagram application and they are then posted without using any native filters.

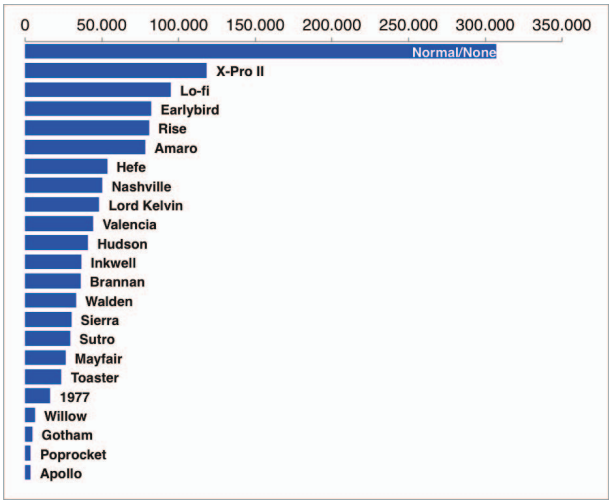


Figure 3. How users apply the filters in their posts.

Some interesting findings emerge on how users interact with the posts that are not posted by themselves. Figure 4 presents the number of likes received by the posts we collected (divided into one-thousand bins up to 10 thousand and above this amount). Similarly to the degree distribution in social networks, the number of likes follows a power-law distribution[11]. Figures 5 and 6 zoom in the first one-thousand bin, showing the likes distribution divided into one-hundred bins and into one-ten bins, respectively. The power-law behavior is preserved. It is worth noting that the behavior is similar in $[0, 10]$ range. Thus, some posts are more attractive than others and they catch more attention. These popular posts are highly likely to be posted by very participative users that tend to attract a large number of followers. In our database, the maximum number of *likes* received by a post was approximately $660K$. The total number of photos that do not receive any like is equal to $622,140$.

[11]Note that the last point is the sum of several one-thousand bins.

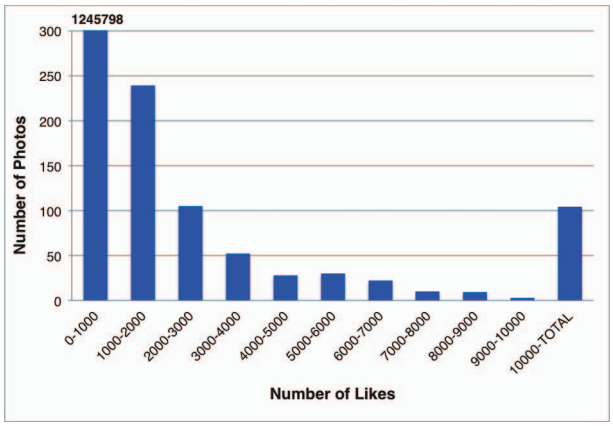


Figure 4. Post likes Distribution. The maximum number of photos with up to one thousand likes is $1,245,798$.

Figure 7 presents the most common tags associated with the posts that received more than 2 thousand likes. Some of them are related to events or celebration dates. For instance, let us focus on the *sochi* tag that appeared several times in our database. We know that the 2014 Winter Olympic Games took place at this Russian city in February and this tag is highly likely to be associated to this worldwide sport event. Another example, is the *valentinesday* tag associated with one of the most important celebration dates in the north hemisphere countries. We have also the *consproject* tag, an event organized by the Converse Company[12].

In Brazil, for instance, we can highlight the *bailedavogue* tag, a big event that occurs during carnival and every year attracts many celebrities and fashion personalities. In this sense, we may suppose that people enjoy interacting with

[12]www.converse.com

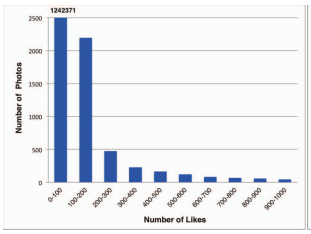


Figure 5. One-hundred bins.

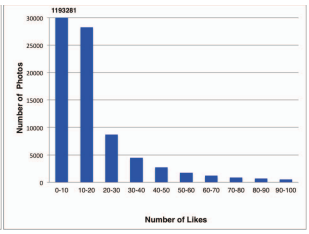


Figure 6. One-ten bins.

posts that are related to current real life events. Moreover, people use Instagram for telling how they are interacting or even reacting to these events.

Figure 7 also shows that posts shared by worldwide brands (*topshop, louivuitton, gucci, disney, converse*) receive a large number of likes. These posts may be shared either by the brands themselves or by their fans. It is well-known that brands use Instagram as a self-promotion media, attracting the public interested in their products. Moreover, worldwide brands are also promoted by bloggers that tend to have a large number of followers increasing the probability of attracting many likes. However, it is worth noting that interaction with posts is not only driven by a large number of followers, but also by the content of the post. We have also language expression tags such as *aw*[13]. Another interesting tag is *regram* which is the act of reposting on Instragram, revealing that people tend to replicate the most interesting posts.

Interestingly is that tag usage could attract likes from people who do not belong to a particular *followed by* list. It means that, tags can attract people who are interested in a Instagram profile in particular, as well as those interested in different types of events (music concerts, fashion weeks, etc.) or photo types (weddings, fashion, fitness, etc.). However, Figure 8 suggests that a large number of tags will not result in more likes. People who received the largest numbers of likes use to associate less than 5 tags in their posts.

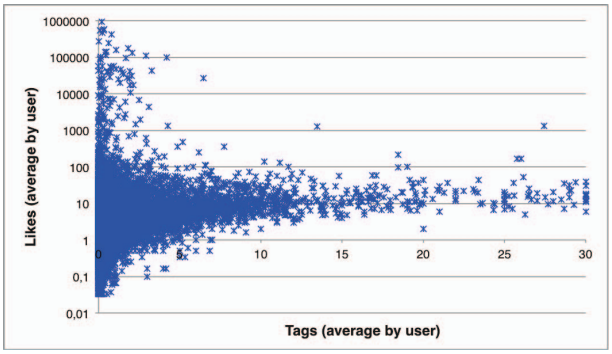


Figure 8. Average of post tags and likes correlation.

Moreover, Figure 9 shows that one feature that influences the number of received likes in an account, as opposed to the number of tags, is just the amount of people who follows the user, inducing the rich get richer phenomenon. Thereby, more followers attract more likes that could turn posts and even tags more popular.

Finally, we turn our attention to how comments and likes are correlated (Figure 10). It is interesting to note that posts with more likes also receive more comments. Users tend to comment posts that were already endorsed by other users,

[13]Used to express, for instance, pleasure, affection.

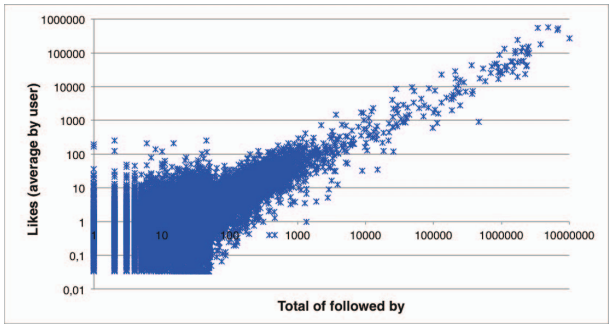


Figure 9. Number of followed by and average of likes correlation.

inducing, again, the rich get richer phenomenon with respect to the post popularity.

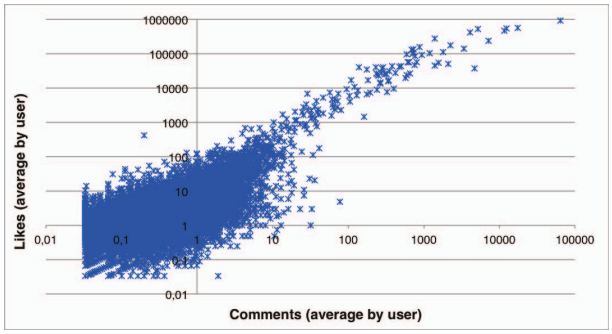


Figure 10. Average of comments and likes correlation.

V. Conclusions

In this work we focus on analyzing how people interact with photographs using the data we collected from the Instagram application. Based on our findings, we have noticed that users tend to concentrate their posts during the weekend and at the end of the day. People also tend to endorse photos with many likes and comments, inducing the rich get richer phenomenon. We could also understand the influence of tags over the amount of likes that some post receives and explore the content of the most popular tags in the data we collected.

We believe that the results presented in this work are the first step to beginning an exploration of how people are interacting with images nowadays. In times when online social networks and mobile technology have been acquiring greater emphasis, we would like to understand the behavior of users *with* photography and mainly *through* photography. One of our research focus is to understand how the social aspects of a tool influence the behavior of users. Furthermore, we want to know if these social aspects could define by themselves a user. Based on some of the results presented in this paper and these research focuses, we will depart to answer some research questions: How to validate our user practices hypotheses applying qualitative analysis? How

Figure 7. Most common posts tags with more than 2 thousand likes.

to perform a meaningful comparison among user practices considering different types of social media? How to extend our analysis applying cultural analytics theory? Is it possible to propose a clustering user practices driven algorithm? We plan to address these questions next.

ACKNOWLEDGMENTS

This research is funded by the authors' individual grants from CNPq, FAPEMIG and the Brazilian National Institute of Science and Technology for Web Research (InWeb).

REFERENCES

[1] N. Hochman and R. Schwartz, "Visualizing instagram: Tracing cultural visual rhythms," in *Proceedings of the International AAAI Conference on Weblogs and Social Media*, 2012.

[2] A. Weilenmann, T. Hillman, and B. Jungselius, "Instagram at the museum: Communicating the museum experience through social photo sharing," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '13. New York, NY, USA: ACM, 2013, pp. 1843–1852.

[3] T. Silva, P. V. de Melo, J. M. Almeida, M. Musolesi, and A. A. F. Loureiro, "You are what you eat (and drink): Identifying cultural boundaries by analyzing food & drink habits in foursquare," in *Proceedings of the ICWSM*, 2014.

[4] A. D. Miller and W. K. Edwards, "Give and take: A study of consumer photo-sharing culture and practice." M. B. Rosson and D. J. Gilmore, Eds. ACM, 2007, pp. 347–356.

[5] R. Chalfen, *Snapshot Versions of Life*. Bowling Green State University Popular Press, 1987.

[6] T. Silva, P. V. de Melo, J. M. Almeida, J. Salles, and A. A. F. Loureiro, "A picture of instagram is worth more than a thousand words: Workload characterization and application," in *Proceedings of the IEEE International CConference on Distributed Computing in Sensor Systems (DCOSS)*, 2013.

[7] N. Hochman and L. Manovich, "Visualizing spatio-temporal social patterns in instagram photos," in *Proceedings of the GeoHCI 2013 Workshop (in conjunction with ACM CHI 2013)*, 2013.

A Rendering-based Method for Selecting the Main Data Region in Web Pages

Leandro Neiva Lopes Figueiredo
Departamento de Computação
Universidade Federal de Ouro Preto
Ouro Preto, MG, Brazil
leomcl@gmail.com

Anderson Almeida Ferreira
Departamento de Computação
Universidade Federal de Ouro Preto
Ouro Preto, MG, Brazil
ferreira@iceb.ufop.br

Guilherme Tavares de Assis
Departamento de Computação
Universidade Federal de Ouro Preto
Ouro Preto, MG, Brazil
gtassis@iceb.ufop.br

***Abstract*—Extracting data from web pages is an important task for several applications, such as comparison shopping and data mining. Much of that data is provided by search result pages, in which each result, called search result record, represents a record from a database. One of the most important steps for extracting such records is identifying, among different data regions from a page, one that contains the records to be extracted. An incorrect identification of this region may lead to an incorrect extraction of the search result records. In this paper, we propose a simple but efficient method that generates path expression to select the main data region from a given page, based on the rendering area information of its elements. The generated path expression may be used by wrappers for extracting the search result records and its data units, reducing its complexity and increasing its accuracy. Experimental results using web pages from several domains show that the method is highly effective.**

***Keywords*-rendering information; visual information; wrapper; main data region; path expression**

I. Introduction

Most of the data available in web pages comes from records stored in databases. In this context, if a website produces several web pages, such pages usually use the same template and are generated by the same script. Generally, such pages are accessed from a search in a website.

These pages, known as search result pages, may contain many data regions, i.e., groups of data in different locations on the screen, for instance, the region that contains the menu, ads, or search result records (SRRs). Each SRR represents an object from a database related with a user search and may contain many attributes. For example, in Fig. 1, the rectangles (a), (b) and (c) are data regions and (d) to (k) are SRRs. Each SRR has attributes, e.g., name, price and user reviews.

Several applications, such as comparison shopping, digital libraries, and data mining and integration, need data from these web pages. Thus, pattern discovery and extraction of data from such pages has required much attention from the scholarly community. Among the steps for extracting data from a search result page, one of the most important is identifying its main data region, i.e., the data region with the SRRs. For instance, on Fig. 1, rectangle (c) is the page's main data region. An incorrect identification may lead to the extraction of irrelevant data (i.e., noises) and, thus, to a decrease on the accuracy of the wrapper.

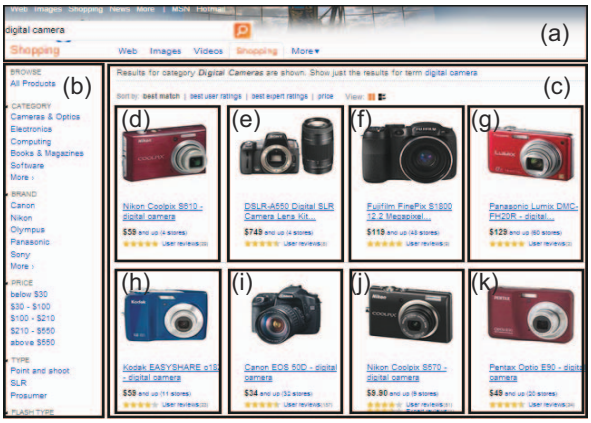


Figure 1. Example of data regions ((a),(b) and (c)), SRRs ((d) to (k)) and data units (e.g., product name, price, user reviews)

In this work, we propose a completely automatic method for identifying the main data region from a HTML web page, analyzing its DOM (Document Object Model) tree and the rendering information of its elements. As a result, the method generates a path expression (i.e., XPath expression) that may be used by algorithms that extract SRRs and its corresponding data units (i.e., attributes).

In web pages, the layout characteristics are known as *rendering area information* and are available when the browser loads the web page (e.g., X and Y coordinates, height and width of each element on the screen). Thus, this information cannot be obtained by analyzing solely the structure of the page.

Like [1] and [2], we assume that the intention of the arrangement of elements on the screen is to emphasize the information the user is looking for, i.e., the region with the SRRs. Thus, this region fills a large area on the screen. This spatial information is obtained when we combine the rendering information (e.g., height and width) of each element. Based on our assumption and the area permeated by the elements, we may decrease the runtime to process

978-1-4799-6953-1/14 $31.00 © 2014 IEEE
DOI 10.1109/LAWeb.2014.14

the DOM tree, focusing only on the most promising regions to contain the SRRs.

Thus, in summary, the main contributions of our work are: (1) completely automatic selection of the main data region, i.e., no human intervention is needed; (2) the method works on a single page, not requiring many page examples to process; (3) it is domain independent, i.e., its rules are not based on specific domains; and (4) the produced path expression may be used by wrappers to extract SRRs and data units.

The rest of this paper is organized as follows: the next section reviews previous related work. Section III describes our method. Section IV reports our experimental evaluation and discusses the results and Section V concludes the paper and discusses future work.

II. Related Work

Information extraction from web pages has been studied in several previous works. Some of them analyze the structure of the pages as a tree, representing hierarchically their elements ([3], [4], [5], [6], [7], [8]). Others use machine learning techniques or probabilistic models to train the algorithm with labeled examples and then induce wrappers to extract data ([9], [10], [11], [12]). Other ones use examples provided by users in a GUI (Graphic User Interface) to find and extract data by analyzing similar objects ([13], [14]) and others use ontologies in the extraction task [15].

The use of structural analysis combined with visual information is found in some works. The VIPS method, proposed in [16], uses the position of the elements in the DOM tree and their visual information (e.g., font name, font color) in order to group similar objects from several segments. Next, it assigns a score to each segment and uses this score to improve search engines, which may use VIPS to detect navigation, decoration, interaction and contact information, and exclude those from the results.

In [17], the authors propose a method that uses positional information of the elements in the DOM along with their visual characteristics (e.g., font style, visibility and rendered area) to find the SRRs and generate candidate XPath expressions to be used in the extraction task. These candidates are ranked by the similarity of their resulting nodes.

Another method, called ViNTs[18], initially uses the visual content (without HTML elements) of the page to identify the regularities from content itself (e.g., text, images, anchors), and then combines those with the HTML tag structure to generate wrappers. The method requires at least five search result pages (from the same web site) with at least four SRRs to identify the regularity among the SRRs that is explored in the wrapper construction. It also requires a special result page called no-result page, with no SRRs, as a negative example.

In [1], the authors represent each page's element as a path expression obtained using its positional and visual information; this path expression is called visual signal. The method aims to identify occurrences of visual signals representing individual data records, parts of data records or sets of data records.

In [19], the proposed method identifies and extracts the main data region based on the visual clue (location of data region, data records and data items on the screen) of web pages. It assumes that the main data region fills the major central portion of the web page and is also a HTML "table" element. As we will demonstrate in Section III, this assertion is not true for the recent web pages, and, thus, other manners to identify the correct main data region are necessary.

The method proposed in [20] represents each element as a symbol and a group of symbols as an alphabet. It analyzes the sequence of symbols of the alphabets to recognize SRRs and, then, the main data region. Afterwards, it excludes all other regions from the source code and uses MDR [21] to read the new structure and extract SRRs and data units. Such method was applied on a collection with 23 search result pages and obtained an accuracy of 86.96% in the task of identifying the main data region.

Some methods assume that SRRs are child elements of the same element ([22], [23]). This assumption may lead to a lower accuracy since the SRRs may have different parent elements. In [24], the same assumption is used and the web pages with child elements of different elements were not considered in the experimental evaluation.

Our method differs from the previous ones since it does not consider the location of the main data region when a browser renders it, does not assume that the elements with SRRs are children of the same element and does not use specific symbols from the HTML language.

III. Proposed Method for Identifying the Main Data Region

With the goal of automatically identifying the main data region of a search result page, we propose a three-step method based on the page rendering information. Our method receives as input a search result page and produces as output an XPath expression to select the main data region. This XPath expression may be used by a wrapper, as initial phase, for extracting the SRRs. The three steps of our method are:

1) The first step generates candidate XPath expressions for selecting the main data region;
2) The second step obtains the visual data based on rendering information; and
3) The third step chooses among the candidate XPath expressions the one that is most likely to select the main data region.

Our method works on the HTML DOM[1] tree, i.e., it manipulates the web page as a tree of objects. In this

[1]http://www.w3.org/DOM/

tree representation, amidst the types of nodes, there are document, element, attribute and text. The *document* node is the root node and has an element node (the element root) as a child. The content of an *element* node, that corresponds to an HTML tag of an HTML page, is the part between the opening and ending tags in the HTML page. The *element* nodes may have elements, attributes and text as child nodes and their child elements are ordered, i.e., we have the first element, second element, and so on. Each *attribute* node has a name and a value attached to an element. And each *text* node has an attached text.

Next, we describe the steps of our method in detail.

A. Determining the candidate XPath expressions

As previously mentioned, the main data region of a search result page (HTML page) contains the SRRs of this page. The main data region may have many SRRs and all SRRs are descendants of the "body" element that is a child of the "html" element (the web page root element). Thus, any other subtree of the "html" element will not be considered by our method (e.g., "head" element).

As first step, our method produces candidate XPath expressions to select the main data region. We consider that all data region with at least a given number of records (i.e., all element with at least a given number of child elements in DOM format) are candidates to be the main data region. We used the $\#_{min}$ parameter to set the minimum number of child elements of a same type that a given parent element must have in order to be considered a candidate. The total number of child nodes of a node does not consider some types of HTML elements that are not used as record delimiters on pages (i.e., elements that indicate form fields, comments, javascript codes, table columns and so on). Thus, our method does not consider elements "input", "textarea", "select", "option", "link", "script", "style", "img", "!–", "td" and "noscript" in the counting process of the total number of child nodes.

After we've selected all nodes that contain at least $\#_{min}$ child elements, our method generates a positional XPath expression for each selected node and adds it to the set of candidate XPath expressions, S. For instance, if our method is executed in a page whose DOM format is shown in Fig. 2, (a) "/html[1]/body[1]/div[1]/ol[1]" and (b) "/html[1]/body[1]/div[2]/ol[1]" are added to S.

In this example, the first expression selects the data region with the page's menu and the second one selects the data region with the page's content containing the SRRs. The final decision regarding the correct XPath expression to select the main data region is obtained in the final of the next two steps.

B. Obtaining the area of each element

In this step, the visual information used by our method corresponds to the attributes (height and width) used to determine the area filled by each element whose expression are in S (i.e., each element selected by the expressions in S) when it is shown in a web browser. In order to obtain such area, our method requires the rendering of the search result page. We use MSHTML[2] to obtain the height and width of each selected data region and then obtain the area.

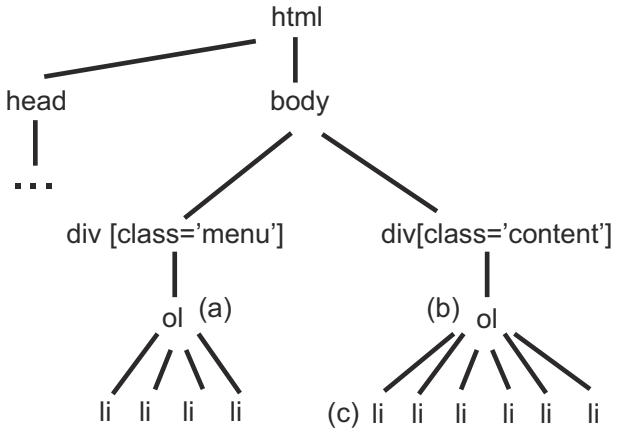


Figure 2. An example of HTML page in DOM format with two data regions (a) and (b), and six SRRs (c).

In detail, the height and width of each element and consequently the area are obtained as follows:

1) First, the web page is loaded by the MSHTML that displays it in the same manner it should be shown by the Internet Explorer browser. When a browser renders a page on the screen, a visual information is associated to each element, such as x and y coordinates and height and width of the element; and
2) Next, our method assigns the corresponding height and width to each element in the DOM tree.

C. Choosing the path expression to select the main data region

The third step of our method must choose, amidst the candidate expressions in S, the one that is most likely to select the main data region. As our candidate expressions are positional expressions, we only have one element selected by each candidate expression, and the previous step has associated to such elements their height and width.

To determine the main data region of a page, our method assumes that this region, which contains the most important information to the user, fills the larger portion of the screen when the browser renders the page.

As formerly stated, the main data region is inside the "body" element of a web page and a browser renders only descendant elements of the "body" element. Thus, to

[2]MSHTML is an dynamic-link (DLL) library that simulates the Internet Explorer browser and may be used by programs written in several visual programming languages.

determine the portion on the screen filled by each element, we calculate the fraction of the area associated to each element regarding the "body" element's area. To determine among the candidate expressions the one that selects the main data region, in this step, our method performs some stages to filter the correct expression in S. Next, we describe each stage.

Filtering the candidate expressions using area

The first stage's aim is to filter, among the candidate expressions, the ones that select an element whose area on the screen figures within the largest ones and, thus, is most likely to contain the SRRs. Formally, let $S = \{exp_1, exp_2, \cdots, exp_n\}$ be the set of n candidate expressions and $E = \{e_1, e_2, \cdots, e_n\}$ be the set of elements selected by each expression in S, where e_i is the element selected by the expression exp_i. Each element e_i and the "body" element have height and width defined in the second step. The area of each element, e_i and "body", is calculated by multiplying its corresponding height and width. We calculate the quota of each area associated to each element e_i ($area(e_i)$) to the area of the "body" element ($area(body)$) and if this quota is bigger than a given threshold $\%_{area}$, the corresponding exp_i is inserted into the set of filtered expressions S', i.e.,

$$S' = \{exp_i \in S | area(e_i)/area(body) \geq \%_{area}\} \quad (1)$$

Filtering the candidate expressions using height

There are elements showed by a browser that fill a large portion on the screen (i.e., a large area), but do not contain SRRs, for instance, the page's header and footer or other elements whose height is very low compared with the "body" element. To solve such problem, we remove from the set S' (i.e., set of filtered expressions) the expressions whose selected elements' heights are too low. If the ratio between the height of an element e_i ($height(e_i)$) and the height of the "body" element ($height(body)$) equals at least a given threshold, $\%_{height}$, its expression exp_i continues belonging to the set S'. Otherwise, the corresponding expression is removed from S', i.e.,

$$S' = S' - \{exp_i \in S' | height(e_i)/height(body) < \%_{height}\} \quad (2)$$

Filtering the candidate expressions using width

Similarly to elements with large areas but low heights, there are also elements whose areas are large but whose widths are short, and do not contain SRRs, for instance, page's side menu. Thus, we also remove from S' the expressions whose corresponding element's width is too short. In order to check whether the width of an element e_i is too short, we divide its width ($width(e_i)$) by the width of the "body" element ($width(body)$) and compare the result with a given threshold $\%_{width}$. If the result is lower than the threshold, we remove its expression from S', i.e.,

$$S' = S' - \{exp_i \in S' | width(e_i)/width(body) < \%_{width}\} \quad (3)$$

Filtering candidate expressions that select ancestral elements

Frequently, a candidate element (i.e., an element selected by a candidate expression) to a main data region contains other candidate elements as children. This situation occurs, for instance, when before a "div"[3] element, that contains the SRRs, there is another "div" element, with a text informing the current shown page (e.g, showing 1 of 10) or yet when after a "div" element there is another "div" element with a panel for changing the page (e.g., "previous page" and "next page"). Thus, both the "div" element containing the SRRs and its parent element may have candidate expressions in S' after the previous filters. It is noticeable that the element with SRRs fills a large area of its parent's area.

In order to remove from S' a candidate expression exp_i that is a prefix of other expression exp_j where the element e_j (selected by exp_j) contains the SRRs, we check whether the ratio between e_j's area and e_i's area is bigger than a given threshold $\%_{parent}$. If the ratio is bigger than $\%_{parent}$, the expression exp_i (the expression that selects the parent element) is removed from S' (the set of candidate expressions), i.e.,

$$S' = S' - \{exp_i \in S' | exp_j \in S' \wedge parent :: e_j = e_i \wedge area(e_j)/area(e_i) > \%_{parent}\} \quad (4)$$

Defining the expression to select the main data region

Among the candidate expressions in S', the one with the largest area will be used to select the main data region, but, this expression may not be able to select the element with all SRRs. This situation occurs because, sometimes, after a browser renders a web page, the SRRs are shown in the same portion on the screen, but, in DOM, the elements that have the SRRs may have sibling elements without SRRs or may have distinct parents, i.e., we cannot consider that elements with SRRs have the same parent element in DOM. An example can be seen in DOM format in Fig. 3. If we apply the previous filters on this DOM, we should have two candidate expressions in S', selecting the "ol" elements in "div" elements (b) and (d). The "ol" element in the "div" element (b) would be considered the main data region since its corresponding area is the largest one. However, this result is incorrect because it does not include all SRRs. As we are using positional expressions, the same

[3]The "div" element defines a division or section in a HTML document.

expression is not able to select both elements. Thus, as an option, we combine the corresponding expressions in only one expression that selects a common ancestral. In Figure 3, the resulting expression would select the "div" element (a).

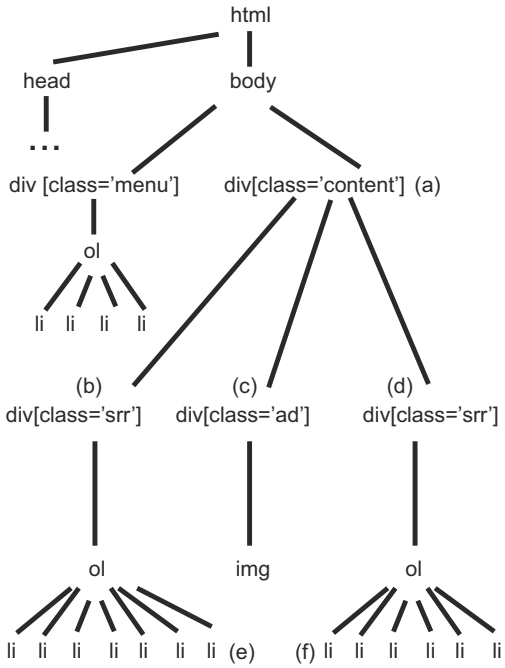


Figure 3. An example of a HTML page in DOM format with two groups of SRRs (e) and (f) in different subtrees (b) and (d). (c) is a region containing advertising and (a) is the main data region.

To solve such situation, our method performs the following stages after the expression exp, with the largest area, and its selected element e, have already been chosen:

1) Let p be the parent of e;
2) If there are descendants of p similar to children of e, p is considered the element with the main data region and a positional expression is generated for it and returned as the expression to select the main data region.
3) If the stage 2 is false, p receives its parent and the stage 2 is performed again.

The method returns to stage 2 at most three times (empirically evaluated). We use the Robust Algorithm for the Tree Edit Distance (RTED) [25] to calculate the similarity between elements (subtrees of DOM). RTED receives as input two elements (subtrees) and returns as output a value related with the number of operations to transform a tree in the other one. For very similar elements, the value is closer to 0. For two identical trees, the value is 0. Our method makes some considerations for using RTED, that are described next.

- For each operation of rename (exchange) of elements, we use weight 3 and, for other operations, we maintain weight at 1.
- To compare two elements (subtrees), we do not use the whole subtree, avoiding the returning of a big value from RTED when a subtree of a SRR with many attributes is compared with another subtree of a SRR with few attributes. We limited the number of levels in the comparison of subtrees to only two levels, i.e., we compare the subtrees considering only the root nodes and the children of the root nodes. Furthermore, as the elements of SRRs are not "br", "h1", "h2", "h3", "h4", "h5", "h6" and "a" elements, that are used to separate visually the SRRs on the screen or show any information related to them, these are ignored.
- As for the final distance between two subtrees that may contain the SRRs, we use the average distance between the possible SRRs in both subtrees. Thus, two subtrees are considered similar if this average distance is, at most, equal to a given threshold δ_{max}.

There are situations in which the set of candidate expressions is empty after the previous filters. In such case, we consider the "body" element as the main data region and our method returns the positional expression "/html[1]/body[1]" as result.

IV. Experimental Evaluation

In this section we present experimental results that demonstrate the effectiveness of our method. We first describe the collections, the baseline and the evaluation metric. Then, we discuss the effectiveness of our method in comparison with the baseline.

A. Collections

In order to evaluate the effectiveness of our method, we adopted collections of web pages used in [17], hereafter referred to as WEBT, and in [20], hereafter referred to as WEBV. The collections are described below.

WEBT collection

The WEBT collection of search result pages was assembled by Trieschnigg et al. [17], combining two collections, Web1 e Web2, and contains 220 web pages. Web1 contains the search result pages obtained from the top 500 US websites listed by Alexa[4], removing websites that require a user account (LinkedIn, Facebook), contain pornographic material and torrent downloads. The authors used the search function on the main page of the webpage for obtaining the search result pages on each website. The complete result page was downloaded in Mozilla Archive Format[5], which includes all images and CSS (cascading style sheet) files and removes javascript. Web2 contains the search result pages obtained from the top UK websites listed by Alexa in a

[4]http://www.alexa.com/topsites
[5]http://maf.mozdev.org/

Table I
DISTRIBUTION OF THE SEARCH RESULT PAGES IN THE APPLICATION DOMAINS.

	WEBV		WEBT	
Domain	# of pages	% of pages	# of pages	% of pages
Academic	3	13.04	3	1.36
Vehicle & Real Estate	1	4.35	4	1.82
Bank	2	8.70	5	2.27
Books	0	0.00	1	0.45
Community	0	0.00	11	5.00
Company	0	0.00	19	8.64
e-Commerce	9	39.13	39	17.73
Game	0	0.00	1	0.45
Government	0	0.00	3	1.36
Images	0	0.00	19	8.64
Job	0	0.00	6	2.73
Music	0	0.00	3	1.36
News	5	21.74	50	22.73
Recipes	0	0.00	1	0.45
Search Engine	2	8.70	26	11.82
Technical Articles	0	0.00	13	5.91
Video	1	4.35	16	7.27
All Domains	23		220	

manner similar to that used for Web1. Websites already in Web1 were removed.

WEBV collection

The WEBV collection was assembled by Velloso and Dorneles [20] and contains 23 search result pages. The search result pages of this collection contains only the source code of the HTML pages, without images and CSS. Thus, its pages are wrongly rendered and present with a messy layout, differently from when we access it on the Web. In order to solve such problem, we accessed the websites again, searched for the same terms as in [20] and saved the whole search result pages on disc, including images and CSS.

Table I shows the popularity of each application domain in both collection.

B. Baseline

We used as baselines the works proposed in [20] and [17], from which we got the collections.

In [20], as described in Section II, the authors attempt to identify the main data region checking sequences of symbols that represent tags in HTML document. Thus, all other regions are removed from the DOM tree and the result is saved as a new page. Next, this new page is used as input for MDR [21] to improve its accuracy, since the number of analyzed nodes has decreased. We check the new pages generated by the method in order to calculate its accuracy. The main data region is correctly identified from a web page, if the new page contains only the main data region from the original page.

In [17], the authors propose a method to generate XPath expressions to extract the SRRs from the pages. In order to use it as baseline, we perform it on each page from the collections and check whether the generated XPath expressions select SRRs from only the main data region. If at least a selected SRR cames from other regions (e.g., menu), we consider the identification of the main data region as incorrect. This method has three parameters: *minSimilarityThreshold*, *avgSimilarityThreshold* and *minimumNodeCount*.

C. Evaluation metric

In order to evaluate the effectiveness our method, we use $accuracy$ metric, calculated as follows:

$$accuracy = \frac{\#correct}{\#total} \quad (5)$$

$\#correct$ is the total number of main data regions from the pages correctly identified by the expression provided by the method in the collection and $\#total$ is the total number of pages in the collection.

D. Experimental setup

Experiments were conducted in each collection. We perform each method on each web page and manually check whether the main data region was correctly selected.

To perform our method, we set $\#_{min}$=3 (the minimum number of child elements), $\%_{area}$=0.1 (the percentage of the area filled by a given element), $\%_{height}$=0.2 (the percentage of the height of a given element), $\%_{width}$=0.3 (the percentage of the width of a given element), δ_{max}=2 (the average distance between subtrees) and $\%_{parent}$=0.2 (the ratio between an element and its parent, when both are candidates). These values are the best parameter settings on WEBT. We make a sensitive analysis to the parameter values in Section IV-E.

To perform the method proposed in [17], we used the same parameter values defined by the authors: *minSimilarityThreshold* = 0.55, *avgSimilarityThreshold* = 0.65 and *minimumNodeCount* = 3. Regarding the method proposed in [20], the only parameter is the minimum size percentage of a region compared with the rest of the sequence, so that a given region can be considered a main data region. This percentage is 0.2.

E. Results

Effectiveness of our method

We have show in Table II the number of main data regions incorrectly selected by our method in WEBV and WEBT collections in each domain. The last line of the table shows the accuracy of our method in each collection.

In the WEBV collection, only 2 main data regions were incorrectly selected by our method. We analyze each case and notice that the first search result page ("bradesco.com.br"), whose main data region was incorrectly selected, uses the "iframe" element to encompass the SRRs. This type of element inserts a page inside another one. Thus, in such a situation, our method is not able to provide an expression to select the main data region, since the "iframe" element appears empty.

Another main data region incorrectly selected by our method is the one in the search result page of the web site

Table II
EFFECTIVENESS OF OUR METHOD IN EACH COLLECTION.

Domain	# of main data regions incorrectly selected WEBV	WEBT
Academic	0	0
Vehicle & Real Estate	0	0
Bank	1	1
Books	0	0
Community	0	0
Company	0	1
e-Commerce	0	3
Game	0	0
Government	0	0
Images	0	2
Job	0	1
Music	0	1
News	1	8
Recipes	0	0
Search Engine	0	1
Technical Articles	0	0
Accuracy	91.30	91.82

"reuters.com". This page contains some elements that are not SRRs, but whose structures are similar to its SRRs. These similar elements induce our method in the last strategy of the third step (defining the expression to select the main data region) to incorrectly construct the XPath expression.

In the WEBT collection, amidst 220 search result pages, we have 18 main data regions incorrectly selected by our method. We checked the corresponding 18 pages and noticed that:

- In 10 search result pages, the failure occurs because our method, in the last strategy of the third step, when trying to find similar subtrees with the subtrees of the SRRs already selected, finds similar subtrees that do not contain SRRs or, due to similarity threshold δ_{min}, subtrees with SRRs are not considered similar;
- In 7 search result pages, the failure occurs due to the fact that elements without SRRs have bigger areas than the element with SRRs, passing through all steps; and
- A search result page, when loaded by the browser, is automatically redirected to another page, and, as the page is different from the page in the collection, we consider incorrect the selection of the main data region.

Effectiveness of our method compared with baselines

Table III shows the comparison of our method with the baselines in both collections, in term of accuracy to select the main data region. In the WEBV collection, the baselines obtained the same accuracy (86.96%). Our method has a gain of around 5% when compared to both baselines. In the WEBT collection, the gains of our method are around 100% and 2.5% compared with the methods proposed in [20] and [17], respectively.

Table III
COMPARISON THE EFFECTIVENESS (ACCURACY) OF OUR METHOD WITH THE BASELINES

Methods	Collections WEBV	WEBT
Our method	91.30	91.81
Proposed method in [20]	86.96	45.45
Proposed method in [17]	86.96	89.55

Table IV
EFFECTIVENESS OF OUR METHOD WITHOUT CONSIDERING A FILTER OR STRATEGY OF THE THIRD STEP.

Filter/Strategy	WEBV	WEBT
Area	86.95	91.81
Height	86.95	87.72
Width	86.95	90.45
Area of parent	0.00	4.55
SRRs in other subtrees	78.26	90.0

Influence of each filter in the final performance

In order to evaluate the influence of each filter/strategy of the third step in the performance of our method, we performed our method without one of the filter/strategies in the last step. Table IV shows the effectiveness (accuracy) of our method in such situations.

We can notice that, in the WEBV collection, when we remove the comparison of the area, height and width of an element selected by a candidate expression with the values of the "body" element, the accuracy of our method drops around 4.8%. When we do not search for SRRs in other subtrees (i.e., subtrees that are not selected by the expression obtained up to such moment) the performance drops around 14.3%. In this collection, the filter that compares the area of a region with the area of its parent leads to the worst performance (our method does not select correctly any main data region).

In the WEBT collection, the filter that compares the area of an element with the "body" element does not alter the final result. When we compare height and width of the elements, as well as the search for similar subtree (SRRs in other subtrees), there is a drop of performance of our method around only 4.5%, 1.5% and 2.0%, respectively. But, as happened in WEBV, if we do not compare the area of an element with its parent's area, our method has a poor performance.

Sensitive to the parameters

We investigated the sensitive of our method to the parameters $\#_{min}$, δ_{max}, $\%_{area}$, $\%_{height}$, $\%_{width}$ and $\%_{parent}$. In each run, we vary the values of only one parameter. Figure 4 shows the accuracy on the WEBT collection when values of $\#_{min}$ and δ_{max} range from 1 to 10. Notice that, there exists a low variation of accuracy when we vary the $\#_{min}$ values. The best accuracies are obtained when we vary the $\#_{min}$ values from 2 to 4. About δ_{max}, the best accuracies are

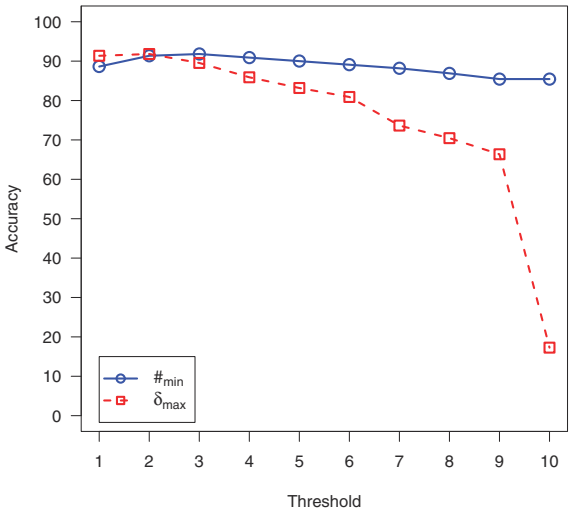


Figure 4. Sensitive to $\#_{min}$ and δ_{max}

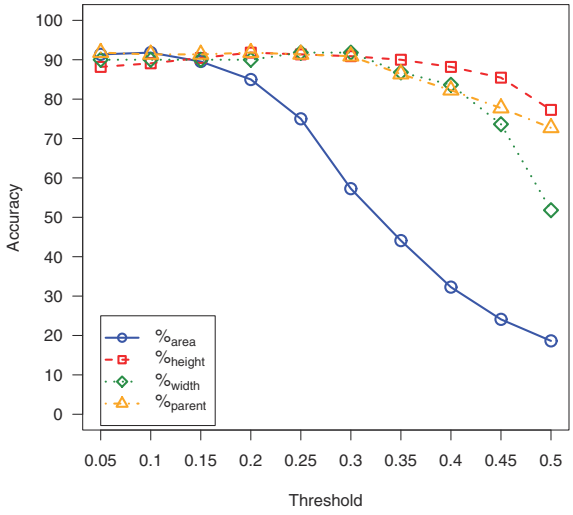


Figure 5. Sensitive to $\%_{area}$, $\%_{height}$, $\%_{width}$ and $\%_{parent}$

obtained when δ_{max}=1 and δ_{max}=2. There exists a slight decrease of accuracy from δ_{max}=3 to δ_{max}=9.

Figure 5 shows the accuracy on the WEBT collection when we vary the values of $\%_{area}$, $\%_{height}$, $\%_{width}$ and $\%_{parent}$ from 0.05 to 0.50. About $\%_{height}$, $\%_{width}$ and $\%_{parent}$, for the values lower than 0.30, our method has the best performances. For $\%_{area} \leq 0.15$, the accuracies are around 90%.

F. Discussion

Although our experiments showed that our method is effective, it has some limitations, which we discuss here.

As any method that works with DOM trees of HTML pages, ours supposes that the source codes of such pages are well structured and have no errors. Although there are many HTML parsers that correct most of these errors, they do not ensure total accuracy. Such characteristic may lead our method to erroneously obtain the height and width of the elements.

Another weakness of our method is due to the incorrect assignment of the rendering information by the browser or MSHTML. For instance, there are situations in which an element filled a large area on the screen but its height and width values are zero. This occurs when we use MSHTML, as well as other web browsers (e.g., Mozilla).

V. Conclusion

In this paper, we propose a method that combines structure and rendering information of HTML pages to generate a positional XPath expression to extract their main data region. Our method uses the height, width and, consequently, the area of each element when a browser renders the page, along with the number of child elements in DOM tree to decide whether an element encompasses the main data region, i.e., the region with the search result records. Our method is unsupervised (i.e., it does not need any examples), domain independent and performs in one page (i.e., it does not need several pages from the same web site to infer how to select the main data region).

We evaluate our method in two collections of search result pages from several domains and compare it with two baselines. The accuracy of our method is higher than 91% and gains are up to 100% when in comparison with the baselines.

As future work, we intend to generate XPath expressions containing predicates to better handle changes on the structure of the pages. Thus, after the method performs on a page from a web site and generates a expression to select the main data region, it will be able to use the same expression to select the main data regions from other pages of the same web site. In this work, when we compare two elements, we compare only their structure (i.e., their subtrees). To improve the method performance, we intend to compare their contents in order to check their similarity. Moreover, we also intend to extend our method for extracting the search result records.

Acknowledgments

This research is partially funded by FAPEMIG (Foundation for Research Support of the State of Minas Gerais).

Moreover, this research was carried out on the Laboratory of Management and Intelligent Analysis of Data of the Federal University of Ouro Preto (GAID/UFOP).

REFERENCES

[1] G. Miao, J. Tatemura, W.-P. Hsiung, A. Sawires, and L. E. Moser, "Extracting data records from the web using tag path clustering," in *Proceedings of the 18th International Conference on World Wide Web*, ser. WWW '09. Madrid, Spain: ACM, 2009, pp. 981–990.

[2] R. Song, H. Liu, J.-R. Wen, and W.-Y. Ma, "Learning block importance models for web pages," in *Proceedings of the 13th International Conference on World Wide Web*, ser. WWW '04. New York, NY, USA: ACM, 2004, pp. 203–211.

[3] A. Sahuguet and F. Azavant, "Wysiwyg web wrapper factory (w4f)," in *Proceedings of the 8th International Conference on World Wide Web*, Toronto, Canada, 1999, pp. 1–22.

[4] L. Liu, C. Pu, and W. Han, "Xwrap: An xml-enabled wrapper construction system for web information sources," *16th International Conference on Data Engineering (ICDE'00)*, vol. 0, p. 611, 2000.

[5] V. Crescenzi, G. Mecca, and P. Merialdo, "Roadrunner: Towards automatic data extraction from large web sites," in *Proceedings of the 27th International Conference on Very Large Data Bases*, ser. VLDB '01. Roma, Italy: Morgan Kaufmann Publishers Inc., 2001, pp. 109–118.

[6] A. Arasu and H. Garcia-Molina, "Extracting structured data from web pages," in *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '03. San Diego, California: ACM, 2003, pp. 337–348.

[7] J. Wang and F. H. Lochovsky, "Data extraction and label assignment for web databases," in *Proceedings of the 12th International Conference on World Wide Web*, ser. WWW '03. Budapest, Hungary: ACM, 2003, pp. 187–196.

[8] Y. Zhai and B. Liu, "Web data extraction based on partial tree alignment," in *Proceedings of the 14th International Conference on World Wide Web*, ser. WWW '05. Chiba, Japan: ACM, 2005, pp. 76–85.

[9] N. Kushmerick, "Wrapper induction for information extraction," Ph.D. dissertation, University of Washington, 1997, aAI9819266.

[10] I. Muslea, S. Minton, and C. Knoblock, "Stalker: Learning extraction rules for semistructured," in *In American Association for Artificial Intelligence (AAAI): Workshop on AI and Information Integration*, Madison, Wisconsin, USA, 1998.

[11] C.-N. Hsu and M.-T. Dung, "Generating finite-state transducers for semi-structured data extraction from the web," *Information Systems*, vol. 23, no. 9, pp. 521–538, 1998.

[12] C.-H. Chang and S.-C. Lui, "Iepad: Information extraction based on pattern discovery," in *Proceedings of the 10th International Conference on World Wide Web*, ser. WWW '01. Hong Kong, Hong Kong: ACM, 2001, pp. 681–688.

[13] B. Adelberg, "Nodose: A tool for semi-automatically extracting structured and semistructured data from text documents," *Special Interest Group on Management of Data - SIGMOD Rec.*, vol. 27, no. 2, pp. 283–294, 1998.

[14] A. H. F. Laender, B. Ribeiro-Neto, and A. S. da Silva, "Debye - data extraction by example," *Data and Knowledge Engineering*, vol. 40, no. 2, pp. 121–154, 2002.

[15] D. W. Embley, D. M. Campbell, Y. S. Jiang, S. W. Liddle, D. W. Lonsdale, Y.-K. Ng, and R. D. Smith, "Conceptual-model-based data extraction from multiple-record web pages," *Data and Knowledge Engineering*, vol. 31, no. 3, pp. 227–251, 1999.

[16] D. Cai, S. Yu, J.-R. Wen, and W.-Y. Ma, "Extracting content structure for web pages based on visual representation," in *Proceedings of the 5th Asia-Pacific Web Conference on Web Technologies and Applications. Xian, China.*, ser. APWeb'03. Xian, China: Springer-Verlag, 2003, pp. 406–417.

[17] R. B. Trieschnigg, K. T. T. E. Tjin-Kam-Jet, and D. Hiemstra, "Ranking xpaths for extracting search result records," Centre for Telematics and Information Technology, University of Twente, Enschede, Technical Report TR-CTIT-12-08, 2012.

[18] H. Zhao, W. Meng, Z. Wu, V. Raghavan, and C. Yu, "Fully automatic wrapper generation for search engines," in *Proceedings of the 14th International Conference on World Wide Web*, ser. WWW '05. Chiba, Japan: ACM, 2005, pp. 66–75.

[19] P. S. Hiremath and S. P. Algur, "Extraction of data from web pages: A vision based approach," *International Journal on Computer Science and Engineering*, vol. 1, no. 3, pp. 50–59, 2009.

[20] R. P. Velloso and C. F. Dorneles, "Automatic web page segmentation and noise removal for structured extraction using tag path sequences." *Journal of Information and Data Management*, vol. 4, no. 3, pp. 173–187, 2013.

[21] B. Liu, R. Grossman, and Y. Zhai, "Mining data records in web pages," in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '03. Washington, D.C.: ACM, 2003, pp. 601–606.

[22] I. V. Jabour, "Impacto de atributos estruturais na identificação de tabelas e listas em documentos html," Mestrado, Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática, 2010.

[23] T. Grigalis, "Towards web-scale structured web data extraction," in *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, ser. WSDM '13. Rome, Italy: ACM, 2013, pp. 753–758.

[24] M. Álvarez, A. Pan, J. Raposo, F. Bellas, and F. Cacheda, "Extracting lists of data records from semi-structured web pages," *Data and Knowledge Engineering*, vol. 64, no. 2, pp. 491–509, 2008.

[25] M. Pawlik and N. Augsten, "Rted: A robust algorithm for the tree edit distance," *Very Large Data Bases (VLDB) Endowment*, vol. 5, no. 4, pp. 334–345, 2011.

Listing Dense Subgraphs in Small Memory

Patricio Pinto, Nataly Cruces and Cecilia Hernández
Department of Computer Science
University of Concepcion
Concepcion, Chile
Email: {patriciopinto,natalycruces,cecihernandez}@udec.cl

***Abstract*—Listing relevant patterns from graphs is becoming increasingly challenging as Web and social graphs are growing in size at a great rate. This scenario requires to process information more efficiently, including the need of processing data that cannot fit in main memory. Typical approaches for processing data using limited main memory include the streaming and external memory models. This paper addresses the problem of listing dense subgraphs from Web and social graphs using little memory.**

We propose an external memory algorithm based on K-way merge-sort for clustering and reordering input graphs. We also propose mining heuristics that work well with different stream orders such as URL, BFS, and cluster-based. Our experimental evaluation shows that on Web graphs, in comparison with the in-memory algorithm, the streaming mining heuristic is able to find between 70 and 96% of edges participating in dense subgraphs, uses only between 17 and 25% of the memory, and running times are between 34 and 65%. We further consider an application that uses these dense subgraphs for compressing Web graphs with a representation that enables querying the collection of subgraphs for pattern recovery and basic statistics without decompression.

***Keywords*-Web Graphs, Graph Pattern Listing, Streaming Algorithms, External Memory Algorithms**

I. Introduction

Discovering patterns from graphs is important for many applications including the Web, social networks, and biological applications among many others. For instance, patterns found in Web graphs and social networks are used to discover link spams and in ranking algorithms for searching the Web. In biological networks, graphs patterns, such as cliques in protein structures are used for modeling and predictions [23]. However, these graphs are growing at an incredible rate. For instance, the Web consists of more than a trillion of pages, increasing in number every day [1]. Social networks are also growing very fast, Facebook is over 1.1 billions active users and twitter was over 500 millions in July, 2013 [2].

At such growth rates comes a need to process that information more efficiently, including the need of processing data that do not fit in main memory. Typical approaches using limited memory include the streaming and external memory models. In the streaming model data is processed sequentially in one or a few passes using limited memory, whereas in the external memory model the idea is to keep data in secondary storage and bring data selectively to main memory to improve I/O performance.

This paper addresses the problem of finding and listing graph patterns based on dense subgraphs from large graphs. We propose an external memory algorithm based on K-way merge-sort for clustering and reordering input graphs. We also propose streaming mining heuristics which work well with different stream orders such as URL, BFS, and cluster-based. We provide experimental evaluation that shows that our external memory algorithm reduces memory usage preserving the quality of the in-memory solution. We also provide a streaming mining heuristic that uses little memory and achieves good quality for stream orders that exploit locality of reference. We further consider an application for compressing Web graphs and social networks that is based on the listed dense subgraphs. The compressed structure not only allows the recovery of the collection of dense subgraphs but also enables basic queries such as obtaining the number and average size of cliques and bicliques as well as other dense subgraphs.

II. Related Work

Finding relevant patterns on graphs have been addressed for some time in different applications. In the context of the Web, Donato et al. [9] used several web mining techniques to discover the structure and evolution of the Web graph, such as weakly and strongly connected components, depth-first and breath-first search. Other proposals use graph algorithms to detect spam farms [22], [12]. Saito et al. [22] present a method for spam detection based on classical graph algorithms such as identification of weak and strong components, maximal clique enumeration and minimum cuts. Gibson et al. [12] propose large dense subgraph extraction as a primitive for spam detection. Their algorithm used efficient heuristics based on the idea of shingles [4].

The streaming model is an important model of computation for processing massive data sets [18], [19]. The most restrictive streaming model allows $O(\text{ polylog } n)$ space and a single or a few passes over the data. To address graph problems in the streaming model, several more relaxed

[1] http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html
[2] http://www.statisticbrain.com/twitter-statistics/

978-1-4799-6953-1/14 $31.00 © 2014 IEEE
DOI 10.1109/LAWeb.2014.16

models have been proposed, such as the semi-streaming, w-stream and stream-sort models. The semi-streaming model allows one or a few sequential read-only passes through the graph, but in addition allows $O(n \text{ polylog } n)$ space, which means that vertices and some information about them can be stored, but not all the edges [10], [19], [27]. The w-stream model is similar, but it allows temporary streams to read/write on disk [21], [27]; and the stream-sort model, which not only creates intermediate streams but also sorts them in a single pass [2], [27].

Feigenbaum et al. [10] propose semi-streaming algorithms for finding approximations to the unweighted maximum dense subgraph matching problem with an approximation ratio $2/3-\epsilon$ in $O(\frac{\log 1/\epsilon}{\epsilon})$ passes using $O(n \log n)$ memory, where $0 < \epsilon < 1/3$. The proposed algorithm is based on first finding a bipartition, then using a matching of the graph, and then finding a set of simultaneous length-3 augmenting paths. The maximum dense subgraph matching algorithm increases the size of the matching by repeatedly finding a set of simultaneously length-3 augmenting paths. They also provide a semi-streaming algorithm for finding a weighted matching. They use edge weights of edges in the stream and compare them with the sum of the weights of the edges in the current matching M. Demetrescu et al. [8] show that the single-source shortest path problem in directed graphs can be solved in the w-stream model in $O(n \log^{3/2} n)/\sqrt{s})$ passes. For undirected connectivity they propose an $O(n \log n)/s$ passes algorithm.

Aggarwal et al. [1] propose a model for dense pattern mining using summarization of graph streams. They define dense patterns based on node-affinity and edge-density of patterns in a general way. Their approach removes small and large adjacency lists a priori because the dense pattern mining definition does not consider them as relevant. On the other hand, running time is in the order of thousand processed edges per second. More recently, Sariyüce et al. [24] propose incremental streaming algorithms for k-core decomposition, where a k-core is defined as a maximal connected subgraph in which every vertex is connected to at least k nodes in the subgraph. The core decomposition of a graph is the problem of finding the set of maximum k-cores of all vertices in the graph. Thus, an algorithm to find k-cores of a graph removes all vertices with degree less than k with their corresponding adjacency edges. The authors propose streaming algorithms supporting insertion and removal of edges for dynamic networks. The algorithms require reordering unprocessed vertices in subgraphs. Stanton and Kliot [25] address the problem of distributed graph partitioning using streaming for directed or undirected graphs. They evaluate different heuristics performed on various stream orders. They find that a greedy linear deterministic algorithm works best together with BFS stream order.

External memory algorithms define memory layouts that are suitable for graph algorithms reducing random accesses to disk. This model has been used for different basic problems, such as scanning, sorting, permuting, and other graph algorithms such as traversal algorithms and graph connectivity[26], [17]. Cheng et al. [7] propose a external memory algorithm for the maximal clique enumeration problem on undirected graphs. The algorithm is based on defining a partition-based strategy that avoids random access. Their algorithm I/O complexity is $O(k \cdot scan(|V+E|)$, $O(kT)$ CPU time, and $O(M)$ memory space, where $k = min\{\frac{|V|(\phi_{deg})^2}{M}, |V|\}$, T is the CPU time complexity of the in-memory algorithm, M is the memory size, and ϕ_{deg} is the maximum vertex degree.

III. Problem Definition

Let $G(V, E)$ a directed graph, with $n = |V|$ vertices and $m = |E|$ edges. We define the neighbors of a vertex v as $adj(v) = \{u : (v, u) \in E\}$. We assume that input graphs use the adjacency list representation, where vertices have unique ids and each adjacency list describes the set of neighbors of a vertex.

Our dense subgraph pattern is described in Definition 1. This definition is based on a complete bipartite graph pattern with an important difference. We add self-loops in each vertex adjacency list, that is, for each vertex v we add edge (v, v) in its adjacency list. This allows us to discover cliques as well as complete bipartite cliques. Similar patterns are defined by Kumar [14], where they only consider complete bipartite graphs.

Definition 1: A *Dense subgraph is based on a bipartite pattern with set overlap* $H(S, C)$ of $G = (V, E)$ is a graph $G'(S \cup C, S \times C)$, where $S, C \subseteq V$.

Note that Definition 1 not only includes cliques ($S = C$) and bicliques ($S \cap C = \emptyset$), but also more general subgraphs.

The problem we want to solve is given a graph G, discover dense subgraphs and list them using small memory. We do not aim for an exact solution, but rather for fast and memory efficient heuristics. We focus on large graphs such as Web and social networks, where $n << m$. These graphs are power-law and present high similarity of adjacency lists and locality of reference. In order to design memory efficient algorithms, we consider the external memory and streaming models. In the context of the external memory model, we use the standard I/O complexity notation [26] in the analysis: M is the main memory size, B is the disk block size, $scan(N) = \Theta(\frac{N}{B})$ I/O, and $sort(N) = \Theta(\frac{N}{B} \log_{\frac{M}{B}} \frac{N}{B})$. To use the streaming model, we consider an input *graph stream* as a stream of adjacency lists as a sequence $X = \{v_0 : adj(v_0), v_1 : adj(v_1), ..., v_n : adj(v_n)\}$.

IV. Heuristics and Algorithms

We first discuss the algorithm that we use as reference (*in-memory*) algorithm (Algorithm 1). We have used it successfully for discovering graph patterns for compressing Web and social graphs [15], [16]. The Algorithm 1 is iterative,

and each iteration consists of a clustering and a mining phase. The clustering phase consists of grouping similar adjacency lists (lists that have similar neighbors) and the mining phase processes the information on each cluster in order to discover the most relevant dense subgraphs, and then extract them from the graph. The next iteration takes as input the remaining graph given by the previous iteration.

The clustering algorithm is based on finding similar adjacency lists using min-hashing similarity [4]. The algorithm represents each adjacency list with P fingerprints (hash values), generating a matrix of fingerprints of $|V|$ rows and P columns. Then it traverses the matrix column-wise. At stage i the matrix rows are sorted lexicographically by their first i column values, and the algorithm groups the rows with the same fingerprints in columns 1 to i, with $0 < i \leq P$. When the number of rows in a group falls below a small number ($threshold$), it is converted into a cluster formed by the vertexes corresponding to the rows. As shown on a previous work, it is sufficient to use $P = 2$ [16] for achieving good results.

During the second phase a mining algorithm is applied on each cluster to discover and extract dense subgraphs. This algorithm first computes frequencies of the nodes mentioned in the adjacency lists of a cluster, and sorts the list by decreasing frequency. Then, each list is inserted into a prefix tree (Definition 2), using a function cost based on the size of dense subgraphs (Definition 3).

Definition 2: We denote a *prefix tree* as $T = (N, A)$, where N is the set of nodes in the tree and $a = (n_x, n_y) \in A$, where $n_x, n_y \in N$ and n_x is the parent of n_y. We define a branch b as the path from the root to a leaf. Each node n_i in a branch has a label and a set S. The label in the node represents a vertex in an adjacency list and the set S consists of all the vertices that share the adjacency list from the root to the n_i in the branch. We define a set C of a node n_i as the set of all the node labels from the root to n_i. The prefix tree allows different nodes in the tree to have the same label.

Definition 3: We consider dense subgraphs whose sizes follow the function $f(T) = max\{|S_i| \cdot |C_i|\}$ on different branches of the tree, with $|S_i| > 1$ and $|C_i| > 1$.

V. External Memory Algorithm

The K-way external merge sort works in two phases [11]. The first is a "run formation" phase, where N input data are streamed in main memory using memory pieces of size M. Each piece of size M is sorted, having at the end of the phase N/M sorted runs. The second phase is the "merge phase", where groups of K runs are merged together. Runs in the merge phase are sorted using buffers of size B. The merge phase might take more than one pass; in each pass one buffer of size B from each run is maintained in main memory and one buffer is used for streaming out sorted runs. Sorting K runs is done using a Heap data structure. Since the memory usage of the algorithm is bound to M and the buffer size is B, $K = \frac{M}{B} - 1$ buffers are used for input and one for output. The overall I/O performance of the algorithm is $O(N/B \log_{M/B} N/M)$, which is a $sort(N)$ primitive in the external memory model.

Algorithm 1 In-memory algorithm for listing dense subgraphs.

Input: G: input graph, P: number of fingerprints, $threshold$: threshold for clustering, $Iters$:iterations, $size\_thr$: minimum size to list ($|S| \cdot |C|$).
Output: Output: $dscol$: Collection of dense subgraphs.

```
1: dscol ← ∅
2: for (i = 1 to Iters) do
3:    Matrix M ← computeFingerprints(G, P)
4:    Clusters C ← getClusters(M, G)
5:    for (c ∈ C ) do
6:       ds ← mine(G, c, size_thr)
7:       dscol.add(ds)
8:    end for
9: end for
```

The external memory algorithm we propose uses the K-way external merge-sort in two different ways: One for sorting the matrix of nP hashes, for the clustering phase and the other for reordering the input graph by cluster id. During the clustering phase, the algorithm computes hashes and sorts them by columns using external merge-sort; clusters ids are defined based on the conditions of pairs of hashes (given that $P = 2$). The vertices ids of each cluster are used for sorting the input graph based on cluster ids. In summary, this external memory algorithm requires two external $sorts$, one over the matrix nP and one for permuting the graph based on vertex id. Therefore, the algorithm is $O(sort(2n) + sort(n + m))$, that is, $O(sort(n + m))$ I/O complexity. This complexity does not consider the mining part of the algorithm.

For the mining phase, the reordered graph is scanned by cluster. We use whatever main memory requires each cluster, which requires at most $O(V_c + E_c)$ time, where V_c is the number of vertices in the cluster and E_c the number of edges. Such external algorithm is given in Algorithm 2.

VI. Streaming Algorithms and Stream Orders

The main idea of our streaming heuristics is to take advantage of the locality of reference found on Web and social graphs. The question we want to answer is whether we can apply a mining algorithm reading a sequential window of neighbors (w) from the input graph stream. We based the heuristic on the mining algorithm we use in the second phase of the *in-memory* algorithm. Our streaming algorithm belongs to the w-stream model. The idea of this model is at each pass one input stream is read, one output stream is written, and data items have to be processed using limited

Algorithm 2 External memory algorithm based on K-way merge sort for listing dense subgraphs.

Input: G, P, $threshold$, $Iters$, $size\_thr$, M, B, K.
Output: Output: $dscol$: Collection of dense subgraphs.

1: $dscol \leftarrow \emptyset$
2: $msFinger.init(M, B, K)$, $msPerm.init(M, B, K)$
3: **for** $(i \leftarrow 1$ **to** $Iters)$ **do**
4: **for** $((v, adj) \in G)$ **do**
5: $fingers \leftarrow computeFingerprints(v, adj, P)$
6: $msFinger.add(v, fingers)$
7: **end for**
8: $sortFingersFile \leftarrow msFinger.extsort()$
9: $cls \leftarrow msFinger.getClusters(sortFingersFile)$
10: $permGpFile \leftarrow msPerm.extperm(M, B, K, cls)$
11: **for** $(cluster \in permGpFile)$ **do**
12: $ds \leftarrow mine(cluster, size\_thr)$
13: $dscol.add(ds)$
14: **end for**
15: **end for**

space. The output stream produced at pass i is the input stream at pass $i + 1$.

Our hypothesis is that if the graph stream has locality of reference, then we can detect different clusters implicitly just sorting the adjacency list by decreasing neighbor frequency and then building a prefix tree every time we find a different frequent first neighbor in any of the adjacency lists. Therefore, after sorting the adjacency lists by decreasing frequency we define a forest of prefix trees, where the root of each prefix tree serves as the identification of clusters in the window. We define a *forest* of prefix trees in Definition 4.

Definition 4: We denote FT a prefix tree forest as a collection of prefix trees, $FT = (r_1, T_1), (r_2, T_2), ..., (r_k, T_k)$, where r_i is the node id of the root of prefix tree T_i, for any $0 < i < k$, with k prefix trees in a window w.

Figure 1 shows an example that illustrates the algorithm. The example shows a window, w, of an input graph. Figure 1-(a) shows the frequency count of each of the neighbors in the adjacency lists showed in Figure 1-(b).

Figure 1-(b) also shows the same graph partition sorted by decreasing neighbor frequency, where we can observe that two clusters are identified. With this information, it is possible to build two prefix trees, where the root of the first is the node 2 and the root of the second is 14 (Figure 1-(c)). Figure 1-(c) also shows the identified dense subgraphs, where the first is $S = (1, 2, 3, 5)$, $C = (2, 3, 4)$, and the second is $S = (11, 12, 13)$, $C = (14, 18, 16)$. Finally, Figure 1-(d) shows our application which compresses the listed dense subgraphs by using a compact representation that is explained in Section VIII. This representation has been previously used for compressing Web and social graphs [15].

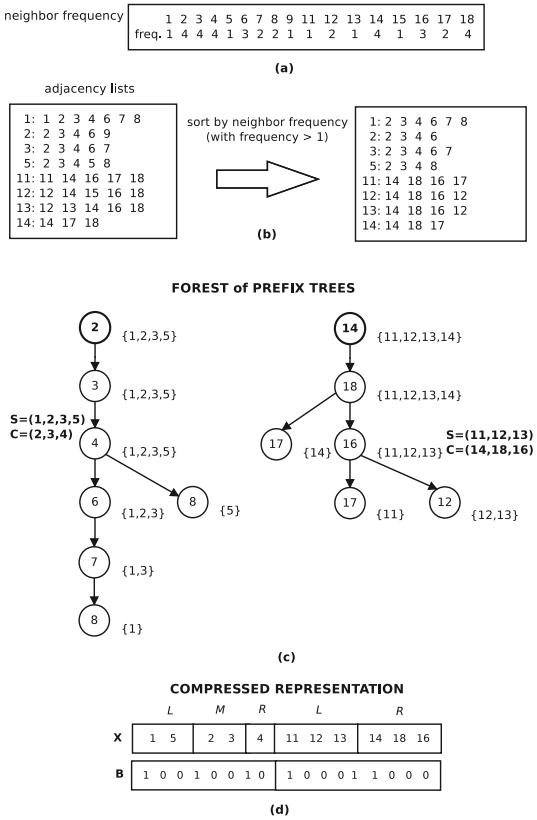


Figure 1. Streaming heuristic example. (a) shows the frequency of the neighbors of the adjacency list showed in (b). (c) shows the forest of prefix trees associated with the adjacency lists sorted by neighbor frequency described in Algorithm 3. (d) shows the compression application which defines the compact representation of the dense subgraphs described in Section VIII.

Algorithm 3 shows the mining algorithm considering a window, w, of edges and a forest of prefix trees.

A. Stream Orders

We consider different stream orders that are good candidates for providing locality [3] and are easy to compute. Specifically, we use the following stream orders:

- URL: URL-based node ordering. In the case of Web graphs, it consists of ordering URLs lexicographically.
- BFS: This node ordering is given by traversing the graph using breadth-first search algorithm starting at a random node.
- CLLP: This node ordering is based on Layered Label Propagation (LLP) clustering described by Boldi et al. [3].
- CHASH: This is not a node ordering in a strict sense, but rather a cluster-based adjacency list ordering. The idea is to group together adjacency lists based on their similarity using the algorithm described in section IV.

Algorithm 3 Streaming algorithm (FT) for listing dense subgraphs.

Input: X:graph stream, w: window of neighbors, $Iters$, $size\_thr$.
Output: Output: $dscol$: Collection of dense subgraphs.

```
1: for (i ← 1 to Iters) do
2:   while (X_i ≠ ∅) do
3:     sortedadjs = readWSortFreq(w, X_i)
4:     Forest FT ← ∅
5:     Tree T
6:     for (adj_u ∈ sortedadjs) do
7:       first ← getFirstElem(adj_u)
8:       T ← FT.find(first)
9:       if (T) then
10:        T.insert(adj_u)
11:      else
12:        T = createT(adj_u)
13:        FT.add(T)
14:      end if
15:    end for
16:    for (T ∈ FT) do
17:      ds ← extractDS(T, size_thr)
18:      dscol.add(ds)
19:    end for
20:  end while
21: end for
```

VII. Experimental Evaluation

We implemented our algorithms in C++. We used a Linux PC with a processor Intel Xeon at 2.4GHz, with 64 GB of RAM and 12 MB of cache. We ran each experiment ten times and considered mean values since the standard deviation was not significant.

We used snapshots of Web graphs and social networks displayed in Table I, which are available by the WebGraph framework project at http://law.dsi.unimi.it. All the algorithms are set to list dense subgraphs with $size\_thr = |S| \cdot |C| \geq 6$ to avoid finding dense subgraphs too small that do not contribute greatly in our compression application. We executed the external memory algorithm described in Algorithm 2, setting M, B, and K so that the sort is done in two passes. We use the URL and CLLP orders provided by the WebGraph project, and compute BFS and CHASH orders. We also consider w of 500, 1000, 2000, and 5000 in the streaming algorithms, since greater values for w did not improve results.

We consider the in-memory algorithm as a reference (shown in Algorithm 1), external memory algorithm (Extmem) (shown in Algorithm 2), the streaming algorithm using only one prefix tree per window (T) and the streaming algorithm considering a forest of prefix trees (FT) described in Algorithm 3.

Dataset	$\|V\|$	$\|E\|$
Eu-2005	862,664	19,235,140
Indochina-2004	7,414,866	194,109,311
Uk-2002	18,520,486	298,113,762
Arabic-2005	22,744,080	639,999,458
Dblp-2011	986,324	6,707,236
LiveJournal-2008	5,363,260	79,023,142

Table I
Main statistics of the Web graphs and social networks.

We examine whether the streaming heuristics are effective for finding and listing dense subgraphs for Web and social graphs. We measure the effectiveness in terms of the number of edges of the graph that participate in dense subgraphs and the required memory and CPU time. We compare streaming algorithms (T) and (FT) using the stream orders presented in section VI-A.

Figure 2 shows how well the streaming algorithms behave in terms of the number of edges participating in dense subgraphs with respect to the time they need to list the dense subgraphs for Web graphs. We observe that there is good locality of reference that allows the FT heuristic to obtain a much better effectiveness that using just one prefix tree (T) per window. We also observe that using a window size $w = 5000$ (right charts) instead of $w = 1000$ allows us to capture more edges in all dense subgraphs using almost the same CPU time. We also observe that the FT algorithm only needs about 5 iterations to capture the dense subgraphs. The best results in terms of edges in dense subgraphs are achieved with CLLP and CHASH orders on Web graphs. Figure 3 shows that in the case of social networks the stream order has more impact than in Web graphs, where the CLLP order exploits locality of reference better than the other stream orders. The figure also shows that social networks need more iterations than Web graphs to list more dense subgraphs.

Second, Table II shows performance ratios in terms of memory usage, total number of edges in dense subgraphs, and CPU time with respect to the in-memory algorithm for 5 iterations on Web graphs and social networks. In addition, Table II shows the Speedup, which is defined as the ratio between the $Edges/secs$ of our streaming algorithm and the $Edges/secs$ of the in-memory algorithm. We compare our results using the FT streaming algorithm over the stream orders URL, BFS, CLLP and CHASH and our external memory algorithm with respect to the in-memory algorithm given in Algorithm 1. *Memory ratio* is the ratio between the amount of memory used by the corresponding algorithm and the in-memory algorithm. *Edge ratio* is the ratio between the number of edges participating in the extracted dense subgraphs obtained by the algorithm and the number of edges achieved using the in-memory algorithm, and *Time ratio* is the ratio between the execution time of the corre-

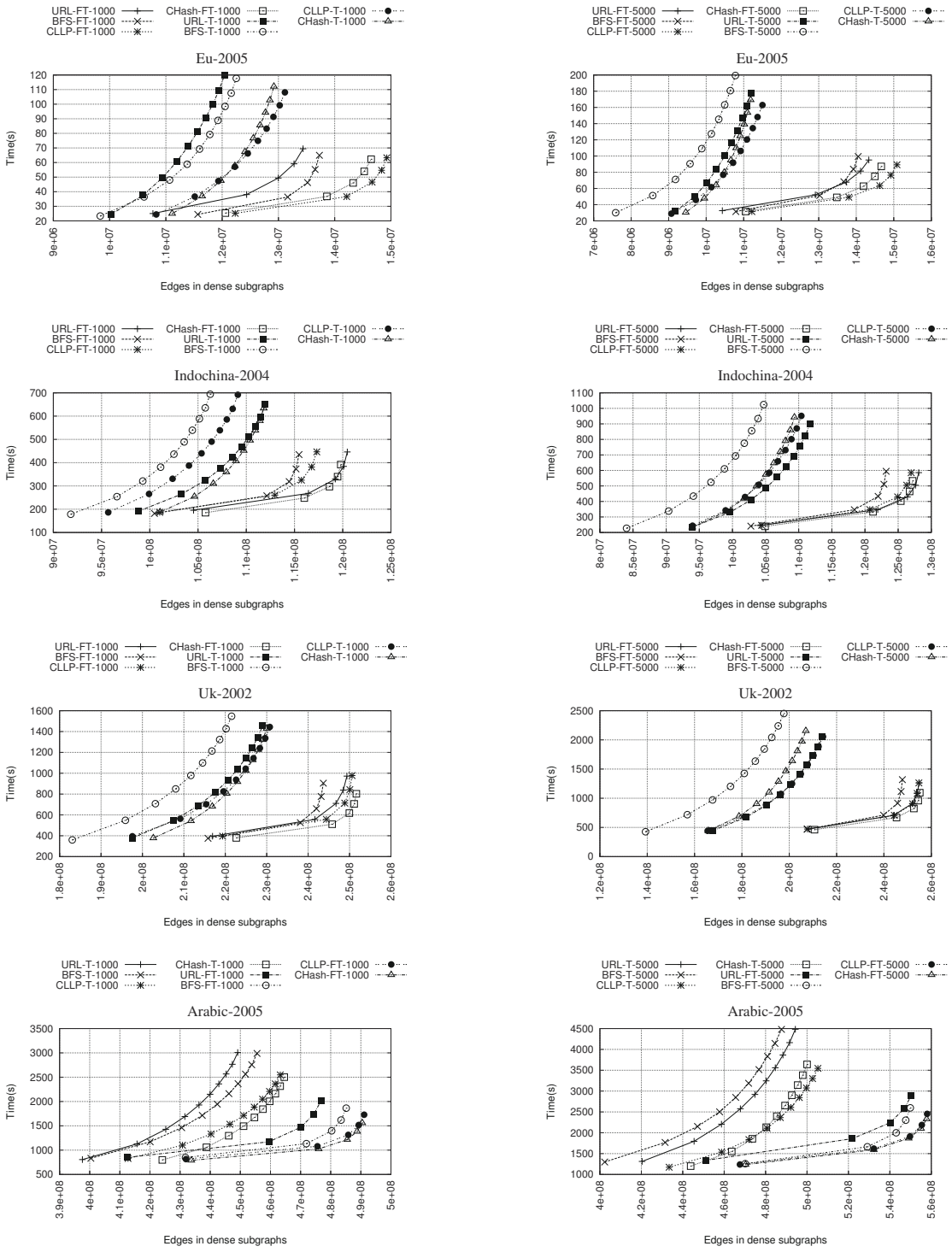


Figure 2. CPU time required for listing edges participating on dense subgraphs in Web Graphs using algorithms T and FT and different stream orders for $w = 1000$ (left charts) and $w = 5000$ (right charts).

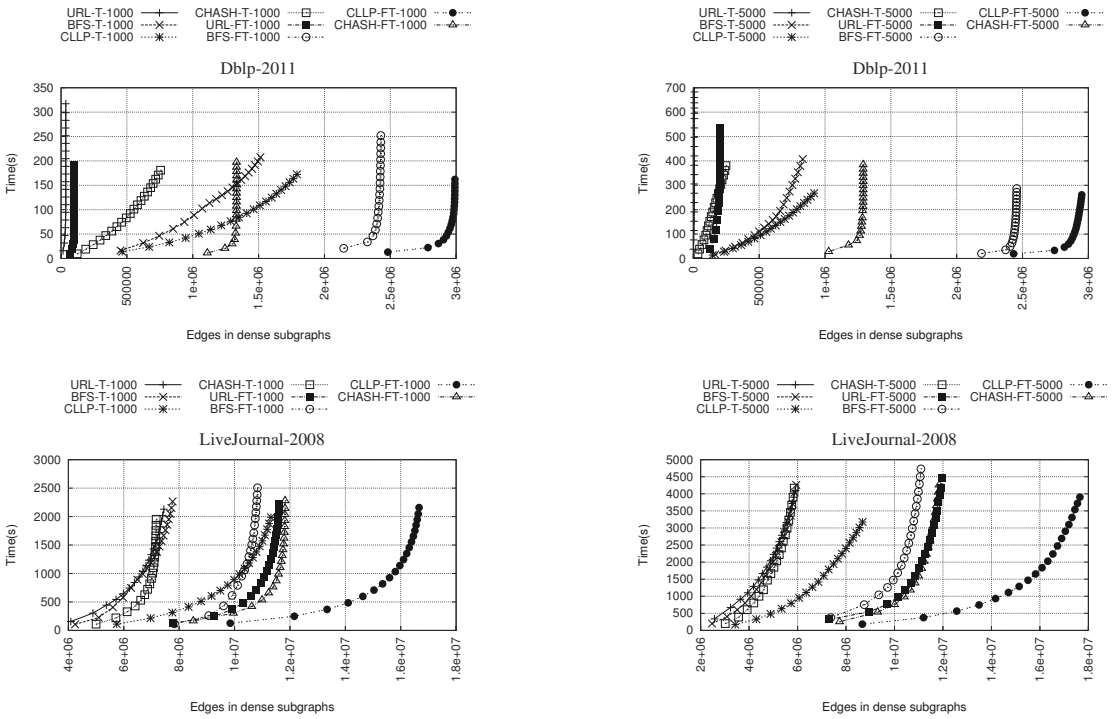


Figure 3. CPU time required for listing edges participating on dense subgraphs in social networks using algorithms T and FT and different stream orders for $w = 1000$ (left charts) and $w = 5000$ (right charts).

sponding algorithm and the execution time of the in-memory algorithm.

We observe in Table II that our external memory algorithm preserves the quality of the in-memory algorithm, using between 53 and 74% of the memory, but doubling running times. Comparing with the in-memory algorithm, we show that for Web graphs, the streaming algorithm (FT) with CLLP order uses between 17 and 25% of the memory between 34 and 65% of the running time, a speedup between 1.21 and 2.15, and achieves between 70 and 96% of the edges in dense subgraphs. However, on social networks the heuristic is less effective being able to retrieve between 40 and 67% of the edges in dense subgraphs, which shows that it pays off to reorder the graph (as in the in-memory algorithm) after each iteration in order to find more subgraphs.

VIII. Application: Compression

The results of listing dense subgraphs can be used for different applications. Here we consider using them for compressing Web graphs and social networks. We represent the collection of dense subgraphs using compact data structures via a symbol sequence and a compressed bitmap. Definitions 5 and 6 describe how dense subgraphs can be used to compress graphs.

Definition 5: Let $G(V, E)$ be a directed graph, and let $H(S_r, C_r)$ be edge-disjoint dense subgraphs of G. Then the corresponding compressed representation of G is $(\mathcal{H}, \mathcal{R})$, where $\mathcal{H} = \{H(S_1, C_1), \ldots, H(S_N, C_N)\}$ and $\mathcal{R} = G - \bigcup H(S_r, C_r)$ is the remaining graph.

Definition 6: Let $\mathcal{H} = \{H_1, \ldots, H_N\}$ be the dense subgraph collection found in the graph. We represent $\mathcal{H}$ as a sequence of integers X with a corresponding bitmap B. Sequence $X = X_1 : X_2 : \ldots : X_N$ represents the sequence of dense subgraphs and bitmap $B = B_1 : B_2 : \ldots B_N$ is used to mark the separation between each subgraph. We now describe how a given X_r and B_r represent the dense subgraph $H_r = H(S_r, C_r)$.

We define X_r and B_r based on the overlapping between the sets S and C. Sequence X_r will have three components: L, Q, and R, written one after the other in this order. Component L lists the elements of $S - C$. Component Q lists the elements of $S \cap C$. Finally, component R lists the elements of $C - S$. Bitmap $B_r = 10^{|L|}10^{|Q|}10^{|R|}$ gives alignment information to determine the limits of the components. In this way, we avoid repeating nodes in the intersection, and have sufficient information to determine all the edges of the dense subgraph. In other words, this

Metric	Order	Datasets					
		Eu-2005	**Indochina-2004**	**Uk-2002**	**Arabic-2005**	**Dblp-2011**	**LiveJournal-2008**
Memory ratio	URL	0.24	0.19	0.19	0.21	0.07	0.12
	BFS	0.21	0.16	0.16	0.19	0.30	0.10
	CLLP	0.25	0.17	0.21	0.24	0.32	0.19
	CHASH	0.26	0.23	0.20	0.26	0.20	0.10
	Extmem	0.59	0.74	0.57	0.53	0.71	0.70
Edge ratio	URL	0.84	0.71	0.96	0.92	0.04	0.30
	BFS	0.82	0.68	0.93	0.92	0.57	0.28
	CLLP	0.88	0.70	0.96	0.94	0.67	0.40
	CHASH	0.86	0.70	0.96	0.94	0.30	0.30
	Extmem	1.00	1.00	1.00	1.00	1.00	1.00
Time ratio	URL	0.69	0.34	0.63	0.70	1.25	0.38
	BFS	0.72	0.35	0.66	0.63	0.49	0.47
	CLLP	0.65	0.34	0.63	0.59	0.46	0.29
	CHASH	0.63	0.31	0.55	0.56	0.72	0.37
	Extmem	2.18	1.82	2.90	2.48	2.40	1.99
Speedup	URL	1.07	2.17	1.48	1.52	0.03	0.79
	BFS	1.01	2.05	1.38	1.70	1.15	0.60
	CLLP	1.21	2.15	1.48	1.83	1.47	1.36
	CHASH	1.20	2.36	1.72	1.92	0.41	0.81

Table II
FT STREAMING HEURISTIC WITH RESPECT TO IN-MEMORY ALGORITHM. $Speedup = \frac{OurAlg.Edges/secs}{InMemoryAlg.Edges/secs}$.

representation allows us to use a sequence X which length is given by $|X| = \sum_r |S_r| + |C_r| - |S_r \cap C_r|$.

Definition 6 describes the compact representation of $\mathcal{H}$ using a symbol sequence and a bitmap. The remaining graph $\mathcal{R}$, in Definition 5, can be compressed using any other compression technique. Here we only show how to compress $\mathcal{H}$ and provide experimental results for that. In order to achieve compression we represent our sequence X and bitmap B using compact data structures. We use Wavelet Trees (WT) [13] using the implementation without pointers [6] for representing the integer sequence X, and compressed bitmaps [20] for the bitmap B. We use the compact data structure, *libcds*, implementation library version 1.0 available at http://www.github.com/fclaude/libcds.

Table III shows some statistics of the listed dense subgraphs using the streaming algorithm FT and CLLP stream order with 5 iterations. We show the distribution of cliques, bicliques and other dense subgraphs with their corresponding average size.

Table IV shows the edge representation in the dense subgraph collection. INMEM shows the percentage of edges captured by the streaming algorithm (FT) using CLLP stream order with respect to the total of edges captured by the in-memory algorithm. RE is the percentage of edges captured by FT with respect to the total number of edges of the graph. Table IV also shows the compression efficiency achieved by the compact representation of the dense subgraphs found using the FT streaming algorithm with 5 iterations. Compression is given by bpe (bits per edge) which corresponds to the space (in bits) of the compact structure of the dense subgraph collection divided by total number of edges in the collection.

Dataset	% CL	size	% BI	size	% DS	size
Eu-2005	4.85	9.95	47.28	22.08	47.85	21.99
Indochina	6.24	5.98	37.61	25.42	56.13	22.94
Uk-2002	3.22	5.31	42.61	20.13	54.16	25.12
Arabic-2005	3.05	5.31	48.06	25.27	48.87	25.83
Dblp-2011	18.27	4.00	11.74	8.30	69.98	6.00
LiveJournal-2008	2.92	3.50	45.44	8.42	51.63	7.95

Table III
PERCENTAGE OF CLIQUES (CL), BICLIQUES (BI), AND THE REST OF DENSE GRAPHS (DS) FOUND, WITH CORRESPONDING AVERAGE SIZE.

Data Set	INMEM (%)	RE (%)	bpe
Eu-2005	88.1	81.0	1.68
Indochina	70.2	66.0	1.26
Uk-2002	96.0	87.2	1.60
Arabic-2005	94.0	88.4	1.27
Dblp-2011	67.6	43.1	5.64
LiveJournal-2008	40.3	18.2	7.95

Table IV
EDGE REPRESENTATION AND COMPRESSION EFFICIENCY IN $\mathcal{H}$.

IX. CONCLUSIONS

This paper proposes an external memory algorithm for finding and listing dense subgraphs in Web and social

graphs. The algorithm preserves the quality of the in-memory algorithm, it reduces memory usage, but double the CPU time. We also present a streaming mining heuristic that takes advantage of the similarity and locality of reference that provide some graph stream orders. We provide experimental evaluation that shows that on Web graphs, in comparison with the in-memory algorithm, the streaming mining heuristic FT using CLLP stream order is able to find between 70 and 96% of edges participating in dense subgraphs, uses only between 17 and 25% of the memory, CPU times are between 34 and 65%, and provides a speedup on $Edges/secs$ between 1.21 and 2.15. However, the results show that on social graphs the streaming algorithm is less effective as we are able to find between 40 and 67% of the edges, using between 19 and 32% of memory, CPU times between 29 and 46%, and a speedup between 1.36 and 1.47. Furthermore, we show an effective way of compressing listed dense subgraphs using compact data structures.

REFERENCES

[1] C. C. Aggarwal, Y. Li, P.S. Yu, R. Jin. *On Dense pattern mining in graph streams*, PVLDB, 2010, 3(1), pp. 975–984.

[2] G. Aggarwal, M. Datar, S. Rajagopalan, and M. Ruhl. *On the streaming model augmented with a sorting primitive*, FOCS, 2004, pp. 540–549.

[3] P. Boldi and M. Rosa, M. Santini, S. Vigna. *Layered label propagation: a multiresolution coordinate-free ordering for compressing social networks*, WWW, 2011, pp.587–596.

[4] A.Z. Broder, M. Charikar, A. M. Frieze, and M. Mitzenmacher. *Min-Wise independent permutations*, J. Comput. Syst. Sci., 60(3), 2000, pp. 630–659.

[5] G. Buehrer and K. Chellapilla. *A scalable pattern mining approach to Web graph compression with communities*, WSDM, 2008, pp. 95–106.

[6] F.Claude and G. Navarro. *Practical rank/select queries over arbitrary sequences*, SPIRE, 2008, pp. 176–187.

[7] J. Cheng, L. Zhu, Y. Ke, and S. Chu. *Fast algorithms for maximal clique enumeration with limited memory*, SIGKDD, 2012, pp. 1240–1248.

[8] C. Demetrescu, and I. Finocchi, and A. Ribichini. *Trading off space for passes in graph streaming problems*, ACM Transactions on Algorithms, 6(1), 2009.

[9] D. Donato, S. Leonardi, S. Millozzi, and P. Tsaparas. *Mining the inner structure of the Web graph*, WebDB, 2005, pp. 145–150.

[10] J. Feigenbaum and S. Kannan and A. McGregor and S. Suri and J. Zhang. *On graph problems in a semi-streaming model*, J. Theor. Comput. Sci., 348(2-3), pp. 207–216, 2005.

[11] H. Garcia-Molina, J. Ullman, and J. Widom. *Database systems - the complete book*, Prentice Hall Press, Upper Saddle River, NJ, USA, 1 edition, 2002.

[12] D. Gibson, R. Kumar, and A. Tomkins. *Discovering large dense subgraphs in massive graphs*, VLDB, 2005, pp. 721–732.

[13] R. Grossi, A. Gupta, and J.S. Vitter. *High-order entropy-compressed text indexes*, SODA, 2003, pp. 841–850.

[14] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. *Trawling the Web for emerging cyber-communities*, Computer Networks, 1999, 31(11-16), pp. 1481–1491.

[15] C. Hernández, and G. Navarro. *Compressed Representation of Web and Social Networks via Dense Subgraphs*, SPIRE, 2012, pp. 264–276.

[16] C. Hernández, and G. Navarro. *Compressed representations for Web and social graphs*, Knowl. Inf. Syst. 40(2), 2014, pp. 279–313.

[17] I. Katriel, and U. Meyer. *Elementary graph algorithms in external memory*, Algorithms for Memory Hierarchies, 2002, pp. 62–84.

[18] I. Munro, and M. Paterson. *Selection and sorting with limited storage*, Theor. Comput. Sci., 1980, (12), pp. 315–323.

[19] S. Muthukrishnan. *Data Streams: Algorithms and applications*, Foundations and Trends in Theoretical Computer Science, (2), 2005.

[20] R. Raman, V. Raman, S.S. Rao. *Succinct indexable dictionaries with applications to encoding k-ary trees and multisets*, SODA, 2002, pp. 233–242.

[21] M. Ruhl, *Efficient algorithms for new computational models*, PhD. Thesis, MIT, 2001.

[22] H. Saito, M. Toyoda, M. Kitsuregawa, and K. Aihara. *A large-scale study of Link spam detection by graph algorithms*, AIRWEB, 2007.

[23] R. Samudrala and J. Moult. *A graph-theoretic algorithm for comparative modeling of protein structure*, Journal of Molecular Biology, 279(1), 1998, pp. 287–302.

[24] A. E. Sariyüce, B. Gedik, G. Jacques-Silva, K. Wu, and Ü. V. Çatalyürek. *Streaming algorithms for k-core decomposition*, PVLDB, 2013, 6(6), pp. 433–444.

[25] I. Stanton, and G. Kliot. *Streaming graph partitioning for large distributed graphs*, SIGKDD, 2012, pp. 1222–1230.

[26] J.C Vitter. *External memory algorithms and data structures*, ACM Comput. Surv., 2(33), 2001, pp. 209–271.

[27] J. Zhang. *A survey on streaming algorithms for massive graphs*, Advances in Database Systems, Springer US, pp. 393–420.

[28] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

Fraud Analysis and Prevention in e-Commerce Transactions

Evandro Caldeira
Federal Center of Technological Education of Minas Gerais (CEFET-MG)
Computing Department
Belo Horizonte, MG, Brazil
Email: evandrocaldeira@gmail.com

Gabriel Brandão
Federal Center of Technological Education of Minas Gerais (CEFET-MG)
Computing Department
Belo Horizonte, MG, Brazil
Email: gabrielbrandao@decom.cefetmg.br

Adriano C. M. Pereira
Federal University of Minas Gerais (UFMG)
Dept. of Computer Science
Belo Horizonte, MG, Brazil
Email: adrianoc@dcc.ufmg.br

***Abstract*—The volume of electronic transactions has raised significantly in last years, mainly due to the popularization of electronic commerce (e-commerce), such as online retailers (e.g., Amazon.com, eBay, AliExpress.com). We also observe a significant increase in the number of fraud cases, resulting in billions of dollars losses each year worldwide. Therefore it is important and necessary to developed and apply techniques that can assist in fraud detection and prevention, which motivates our research. This work aims to apply and evaluate computational intelligence techniques (e.g., data mining and machine learning) to identify fraud in electronic transactions, more specifically in credit card operations performed by Web payment gateways. In order to evaluate the techniques, we apply and evaluate them in an actual dataset of the most popular Brazilian electronic payment service. Our results show good performance in fraud detection, presenting gains up to 43 percent of an economic metric, when compared to the actual scenario of the company.**

***Keywords*-Fraud Prevention; e-Commerce; e-Business; e-Payment; Machine Learning;**

I. Introduction

Recently we have observed a significant increase in the volume of electronic transactions, mainly due to the popularization of World Wide Web and electronic commerce, such as online retailers (e.g., www.ebay.com, www.walmart.com, www.amazon.com). We also testify a huge increase in the number of online frauds, resulting in billions of dollars losses each year worldwide. Therefore it is important and necessary to developed and apply techniques that can assist in fraud detection, which motivates our research.

Bhatla et al [1] said that the rate at which Internet credit card fraud occurs is 12 to 15 times higher than face-to-face transactions. The 12th annual online fraud report by CyberSource [2] shows that, for most of the current decade, merchant online fraud losses continued to increase, reaching a peak of $4 billion in 2008. According to Siddhartha Bhattacharyya et al. [3] with the growth in credit card transactions, as a share of the payment system, there has also been an increase in credit card fraud, and 70% of U.S. consumers are noted to be significantly concerned about identity fraud.

Moreover, many fraud detection problems occur in huge amounts of data. For instance, the credit card company Barclaycard has about 350 million transactions per year just in the UK. The Royal Bank of Scotland, which has the largest credit card market in Europe, has more than one billion transactions per year [4]. The processing of these datasets, looking for fraudulent operations, requires fast and efficient algorithms.

In this context, data mining techniques have been relevant in solving this challenge since it can deal with a large amount of data. In this work we apply and evaluate computational intelligence techniques to identify fraud in electronic transactions, more specifically in credit card operations. In order to evaluate the techniques, we define a concept of economic efficiency and apply them in an actual dataset of the most popular Brazilian electronic payment service. Our results can be used to create systems to assist fraud analysts in their jobs. The performance in fraud detection, compared with the actual scenario, is up to 43% improvement in the financial gain, that is, using the economic efficiency metric that will be later explained.

The remainder of this paper is organized as follows. Section II describes some related work. Section III presents a brief description about the computational intelligence techniques that we adopt in this work: Bayesian networks, logistic regression, neural networks and random forest. Section IV describes our case study, using a representative sample of actual data, where we present a dataset overview, the experimental methodology and results. Finally, section V presents the conclusions and future work.

II. Related Work

Due to the importance of the fraud detection problem, we may distinguish several works that discuss this subject[3], [5], [6], [7]. Thomas et al. (2004) [8] propose a very simple decision tree that is used to identify general fraud classes. They also propose a first step towards a fraud taxonomy. Vasiu and Vasiu (2004) [9] propose a taxonomy for computer fraud and, to build it, employ a five-phase methodology. According to the authors, the taxonomy presented was prepared from a fraud preventing perspective and may be used in various ways. For them, this methodology can be useful as a tool for awareness and education, and can also

978-1-4799-6953-1/14 $31.00 © 2014 IEEE
DOI 10.1109/LAWeb.2014.23

help those responsible for combating frauds associated with IT to design and implement policies to reduce risks. Chau et al. (2006) [10] propose a methodology called *2-Level Fraud Spotting (2LFS)* to model the techniques that fraudsters often use to carry out fraudulent activities and to detect offenders preventively. This methodology is used to characterize the auction users on-line as honest, dishonest, and accomplices. Methodologies that characterize fraud are essential for the first phase of the process, since they are the starting point to create a model of the problem and define the best technique for its solution.

There are several researches that develop methods to detect fraud [11], [5], [12] and we can realize that these methodologies can differ significantly due to the peculiarities of each fraud type. However, what can be noticed is that the data mining techniques have been widely used in fraud detection regardless of the methodology adopted. This is because these techniques allow the useful information extraction in databases with large volumes of data. Phua et al. [13] conducted an exploratory study of numerous articles related to fraud detection using data mining and explained these methods and techniques.These algorithms are based on some approaches such as supervised strategy with labeled data, unsupervised strategy with unlabeled data and hybrid approach.

In supervised strategy with labeled data, algorithms examine every transaction, previously labeled, to mathematically determine the profile of a fraudulent transaction and estimate your risk. Neural Networks, Support Vector Machines (SVM), Decision Trees and Bayesian Networks are some of the techniques used by this strategy. Maes et al. [14] used the STAGE algorithm for Bayesian networks and "back propagation" algorithm for neural networks to detect fraud in credit card transactions. The results show that Bayesian networks are more accurate and faster training, but are slower when applied to new instances.

In unsupervised strategy with unlabeled data, the methods do not require prior knowledge of fraudulent and not fraudulent transactions. On the other hand, changes in behavior are detected or unusual transactions are identified. Examples of these techniques are Clustering and Anomaly Detection. Netmap [15] describes how the clustering algorithm is used to form well-connected data groups and how it led to the capture of the real insurance fraudsters. Bolton and Hand [16] proposed an approach of fraud detection for credit card using anomalies detected in transactions. Abnormal behaviors are identified in spending and how often they occur is used to determine which cases may be fraud.

In the hybrid approach (supervised and unsupervised) there are researches using data labeled with supervised and unsupervised algorithms to detect fraud in insurance and telecommunications. Unsupervised approaches have been used to segment data into groups to be used in supervised approaches. Williams and Huang [17] apply a three step process: k-means for detecting groups, C4.5 for decision making, and statistical summaries and visualization tools to evaluate the rule. It is important to note that the choice of which approach to be used depends on the methodology and the available database.

SVM and random forests are sophisticated data mining techniques, which have been noted in recent years to show superior performance across different applications [18], [19] SVMs are statistical learning techniques, with strong theoretical foundation and successful application in a range of problems [20]. They are closely related to neural networks, and through use of kernel functions, can be considered an alternate way to obtain neural network classifiers. Rather than minimizing empirical error on training data, SVMs seek to minimize an upper bound on the generalization error. As compared with techniques like neural networks which are prone to local minima, overfitting and noise, SVMs can obtain global solutions with good generalization error. Appropriate parameter selection is, however, important to obtain good results with SVM. In our application, which has a very unbalanced data, SVM does not provide good results.

There is a very complete work [21] that performs a review of the literature on the application of data mining techniques for the detection of financial fraud. Although financial fraud detection (FFD) is an emerging topic of great importance, a comprehensive literature review of the subject has yet to be carried out. This paper thus represents the first systematic, identifiable and comprehensive academic literature review of the data mining techniques that have been applied to FFD. 49 journal articles on the subject published between 1997 and 2008 were analyzed and classified into four categories of financial fraud (bank fraud, insurance fraud, securities and commodities fraud, and other related financial fraud) and six classes of data mining techniques (classification, regression, clustering, prediction, outlier detection, and visualization). The findings of this review clearly show that data mining techniques have been applied most extensively to the detection of insurance fraud, although corporate fraud and credit card fraud have also attracted a great deal of attention in recent years. The main data mining techniques used for FFD are logistic models, neural networks, the Bayesian belief network, and decision trees, all of which provide primary solutions to the problems inherent in the detection and classification of fraudulent data. This paper also addresses the gaps between FFD and the needs of the industry to encourage additional research on neglected topics, and concludes with several suggestions for further FFD research.

These related works have helped us, indicating promising strategies for detecting and preventing fraud. As the datasets are different, mainly due to the very unbalanced data of our scenario, it is not possible to directly compare the results, but they provide an idea of the efficiency of these approaches.

Moreover, as in our case the main goal is to rank the transactions to block the ones that have high probability of being fraud (chargeback), we are going to define a more precise quality indicator to measure the economic gain of each computational model.

III. Fundamentals

This section describes the techniques we apply and evaluate in this work: Bayesian networks (Section III-A), logistic regression (Section III-B), neural networks (Section III-C), and random forest (Section III-D).

A. Bayesian Networks

Bayesian Networks (BN) are directed acyclic graphs that represent dependencies between the variables of a probabilistic model, where each node in the graph represents a random variable and the arcs represents the relationships between these variables [22], as showed by Figure 1, where the event A affects directly the event D that if affected directly by event B, and so on. And *e* is an independent event.

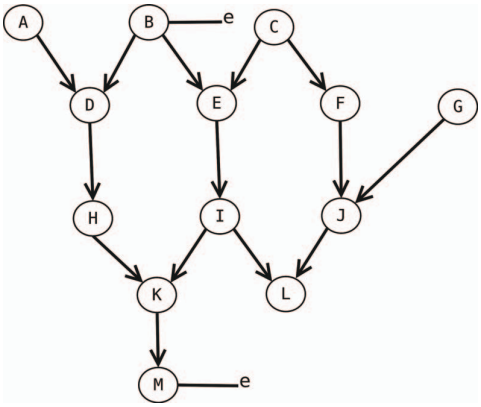


Figure 1. Bayesian Network - Description.

The mathematical definition for BN is derived of Bayes theorem, which shows that conditional probability of a event A_i given a event B, can be calculated by Equation 1, where $P(A_i|B)$ is the probability of A when B occurs.

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{P(B)} \tag{1}$$

In fraud detection problem the BN is unknown, therefore to build the BN graph it is need to learn it from the data. From the BN graph, we can calculate the set of dependent variables to happen a fraud (conditional probability), using Equation 1. Before calculating the conditional probability, we can find the probability of fraud applying Equation 2 [23].

$$P(x_i, ..., x_n) = \prod_{i=0}^{n} P(x_i|Parents(X_i)), \tag{2}$$

where $Parents(X_i)$ are determined by a graph as showed by Figure 1.

B. Logistic Regression

Logistic Regression (LR) is a statistical technique that produces, from set of explanatory variables, a model that can predict values taken by a categorical dependent variable. Thus, a regression model is used to calculate the probability of an event, through the *link* function described by the following Equation:

$$\pi(x) = \frac{e^{(\beta_0+\beta_1 x_1+\beta_2 x_2+...\beta_i x_i)}}{1+e^{(\beta_0+\beta_1 x_1+\beta_2 x_2+...\beta_i x_i)}}, \tag{3}$$

where $\pi(x)$ is the probability of success when the value of the predictive variable is x. β_0 is a constant used for adjustment and β_i are the coefficients of the predictive variables [24].

In order understand LR, it is important to explain the concept of *Generalized Linear Models* (GLM). This consists of three components [25]:

- A random component, which contains the probability distribution of the dependent variable (Y).
- A systematic component, which corresponds to a linear function between the independent variables.
- A *link* function, that is responsible for describing the mathematical relationship between the systematic component and random component.

The binary LR model is a special case of the GLM model with the *logit* function. This function is used to get the estimation of coefficients [26]. Then, we apply these coefficients in Equation 3 that result in our fraud probability.

C. Neural Networks

A Neural Network (NN) is an interconnected assembly of simple processing elements, units or nodes, whose functionality is loosely based on the animal neuron [27]. The processing ability of the network is stored in the inter-unit connection strengths, or weights, obtained by a process of adaptation to, or learning from, a set of training patterns.

Generically, the processing in a neuron consists of a linear combination of entries (x_j), which can be described by Equation 4:

$$\begin{aligned} net &= w_1 * x_1 + w_2 * x_2 + ... + w_D * x_D \\ &= \sum_{j=1}^{D} w_j x_j = \underline{w}^T * \underline{x}, \end{aligned} \tag{4}$$

where w_j is a weight associated with the input (x_j). This weight shows the intensity wherewith a particular input influences the output value. The calculated value (net) is applied in an activation function that can be Linear, Step, Ramp, Sigmoid, Hyperbolic Tangent or Gaussian. [28] The NN model used was MultiLayer Perceptron (MLP), which has the ability to classify non-linearly separable regions [29], appropriate for our fraud detection approach.

The training was done using the Levenberg-Marquardt algorithm [30], because it is fast and can achieve good results. We perform a set of experiments to determine the best NN configuration, that is, a network with two layers: the first (hidden layer) containing ten neurons and the second (output layer) containing one neuron.

D. Random Forest

The Random Forest (RF) algorithm was proposed by Breiman [31] based on the use of trees to product classification. Breiman's definition to algorithm is: "A RF is a classifier consisting of a collection of tree-structured classifiers $h(x, \theta_k), k = 1, ...$ where the θ_k are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x".

The classifier quality or performance can be measured by a high value of probability $\mathcal{P}(h(X) = Y)$. The vector X represents the variables of the problem and Y is the response. Given a observed dataset

$$((x_{1,1}, ...x_{1,n}), (x_{2,1}, ...x_{2,n}), ..., (x_{k,1}, ...x_{k,n})) = D$$

and let B be the number of trees and m the number of features. The Algorithm 1 describes the RF.

Algorithm 1 Random Forest Algorithm

```
for N = 0, .., B do
    D_i ← Bootstrap sample from D
    T_i ← Contruct tree using D_i
    for node = 1, .., No.Nodes do
        node_i ← choose random subset m of all features.
    end for
end for
X ← take the majority vote for all trees
```

IV. Case Study

This section presents our case study where we apply the computational intelligence techniques to detect fraud in electronic transactions, more specifically in credit card in terms of chargeback operations.

A. DATASET OVERVIEW

PagSeguro[1] is a Web service for online payment, owned by the largest Latin America Internet and Web Content Provider, named Universo Online Inc.(UOL)[2], which ensures the safety of those who buy and sell on the web.

In *PagSeguro* each transaction is composed of tens of attributes of the more different types and one of these attributes refers to the status of the transaction, which can result in a valid transaction or chargeback. The purpose of this work is to analyze a set of transactions that occurred in *PagSeguro*, using the attributes that characterize these transactions to apply computational intelligence techniques, such as Bayesian Networks, Logistic Regression, Random Forest and Neural Networks, to detect fraud (chargeback).

Table I shows a short summary of the *PagSeguro* dataset. It embeds a significant sample of valid and chargeback transactions, which has thousands of transactions. Due to a confidentiality agreement, the quantitative information about this dataset cannot be presented.

	Valid	Chargeback
Average Value (US$)	36.33	81.59
Standard deviation (US$)	80.51	122.74
Median (US$)	15.00	40.00
Coefficient Of Variation	2.22	1.50

Table I
PagSeguro Dataset - Summary.

Figure 2 shows the relative quantity of chargeback transactions for each month. Despite this percentage would be considered low, it is very significant, since a chargeback transaction results in a loss of the total transaction value. Moreover, a valid transaction results in a gain of only a small percentage of the transaction value for the payment service company.

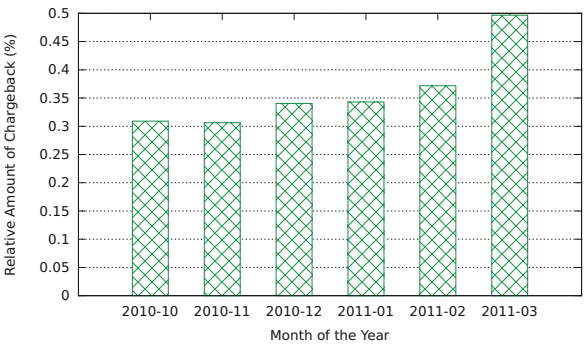


Figure 2. Relative Amount of Chargeback.

Figure 3 shows the cumulative distribution function (CDF) of transaction value. Valid transactions with values lower than US$25 correspond to 66%, and 32% for chargebacks ones. Thus, we can see that in general valid transactions present lower values than chargeback ones.

From the dataset we selected 21 attributes to be used as candidate for the techniques. The most important attributes that we use are described, as follows:

- **Value**: a numeric attribute that represents the value of transaction.
- **Score**: a literal attribute that helps to identify successful, unsuccessful and incomplete transactions.
- **Hour**: a numeric attribute that refers to the transaction create time.

[1]http://pagseguro.uol.com.br
[2]http://www.uol.com.br

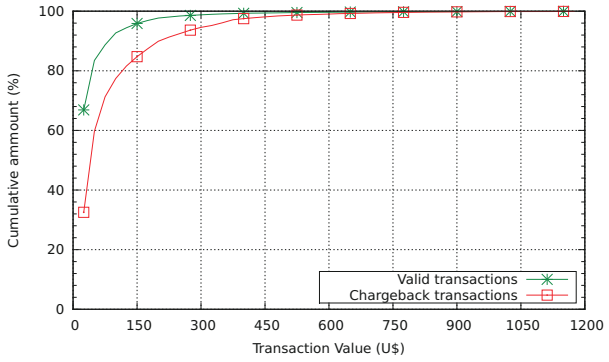


Figure 3. Cumulative Distribution Function (CDF) of Transaction Value.

- **Buyer Type**: a literal attribute that helps to identify a type of user buyer type.
- **Buyer Registration Time**: a numeric attribute that display the buyer's registration time.
- **Day**: a numeric attribute that display the day in which the transaction occurs.
- **Buyer's Points**: a numeric attribute that helps to identify users who had successfull transactions in the past.
- **Registered** $_{\text{flg\_registered}}$: A Flag attribute. In PagSeguro, users unregistered also can buy, this flag helps to identify who are the registered buyers.
- **Seller Registration Time**: a numeric attribute that display the seller's registration time.
- **Stores Main Category**: a numeric attribute that is related to the main category of the store. The category refers to the main products type sold by the store.
- **Credit Card Operator**: a numeric attribute that identify credit card operator.
- **Credit Card Owner Age**: a numeric attribute that represents, in years, how old is the credit card owner.
- **Quantity of installments** $_{\text{num\_installment\_qty}}$: A numeric attribute that it says the quantity of installments used in the purchase.
- **Status at the *Serasa***[3] $_{\text{idt\_serasa\_status}}$: A literal attribute that it shows the status of the buyer at the Serasa .
- **Had response from *Serasa*** $_{\text{flg\_has\_answer\_sr}}$: A flag attribute that it shows if the consults at the Serasa returns an answer about the buyer.
- ***CPF***[4] $_{\text{flg\_cpf}}$: A flag attribute. This attributte tells that CPF of the buyer is the same of the CPF of credit card owner.

[3]The Serasa is a private company that owns one of largest data base in the world and devotes its activity to the provision of services of general interest. The institution is recognized by the code of consumer protection as a entity of public nature.

[4]A document that identify the individual taxpayer in face of the Federal Revenue Secretariat of Brazil (FRSB). The CPF holds the registration information provided by the individual taxpayer that the other data systems of the FRSB.

- **DDD**[3]: a flag attribute that compares if the DDD of the registered user in the *PagSeguro*is consistent with the DDD of the credit card owner.
- **Federation Unit**: a nominal attribute that refers to the Federation Unit provided by the user.

B. Methodology

We used the same methodology for all techniques, starting with a characterization of our dataset, which allowed us to remove items with lower significance and categorize some numeric variables. We made a selection of the most relevant attributes to fraud detection, using "Forward Stepwise Regression", which is based in the verisimilitude concept [32]. We also use InfoGain, which shows the relative gain of each variable, and this was made in Weka[4]. Weka is a free software, under GPL License and it has many data mining and classification algorithms in its toolbox.

Following this process we define the training and test sets to evaluate the algorithms. We use the first 3 weeks of the month for training and remaining for test. This reproduces a real scenario situation and guarantees the model generality. We also use the technique of "K-fold-Cross-Validation" to validate the quality of our experiments. In order to perform this, we define the sub-samples number (K) to 5.

To evaluate the fraud detection techniques we use different environments, each of them has its own parameters. The fine tuning of these parameters was made using an exhaustive search testing different values for each technique. Next, we describe some details about the techniques and experiments, such as the parameters used for each technique.

- We use the software R[5] to build the LR model. To binary LR we use GLM package with the parameters: "FORMULA" where we set the response variable to chargeback and independent variables the others. "FAMILY" is defined as binomial and "LINK" as logit.
- For the BN we use Weka. We use the parameter **Q** to algorithm Hill Climbing to search for the network topology, as subcategory of it we have the parameters: "-P" as the maximum number of father nodes set to 1 because this the number of our response variable and "-S" to define the score that is used to mount the Bayes probability table.
- For the NN we use MATLAB Neural Network Toolbox. The network is a MLP with one hidden layer consisting of ten neurons and with the output layer of one neuron. The activation function of the hidden layer is tangent sigmoidal and linear for the output layer. For the training stage, we use the Levenberg-Marquardt algorithm.
- For the RF we use Weka. This implementation only permits the manipulation of the "Max Depth" of the

[3]Long Distance Call

[4]http://www.cs.waikato.ac.nz/ml/weka/

[5]http://www.r-project.org/

trees, the "Number of Features" to be used in the random selection and the "Number of Trees". We set "Max Depth" to unlimited as well as the "Number of Features" and "Number of Trees" to 10.

After the execution of the techniques we construct a ranking by the degree of reliability to fraud assigned to each transaction. On the top of the ranking would be the transactions with the higher probability to be fraud. After this, we apply Equation 7 in many ranking ranges to obtain the best result.

Beyond the precision we measured the recall that is the ability to find all the existent frauds. We also defined a fitness function called Economic Efficiency (EE) that can be seen on Equation 5. The Gain (G) represents the financial value of true positive transactions, rate (r) is a percentage that the company gains in a successful transaction and the Lost (L) is the financial value of false negative transactions. Applying this formula in the ranking we find the position that maximizes the profit for a given algorithm.

$$EE_{Technique} = \sum_{j=1}^{n} G_j \times r - L_j \times (1 - r) \quad (5)$$

Equation 6 is a simplification of the Company profit. The **r** value has the same meaning as described before in Equation 5, **NF** represents the financial value of Non-Fraud transactions and **F** is the financial value of fraud transactions.

$$EE_{Real} = \sum_{j=1}^{n} NF_j \times r - F_j \times (1 - r) \quad (6)$$

Equation 7 gives a relative gain where 100% represents the maximum gain and 0% is the actual scenario without the use of any technique. The EE_{Max} is the maximum gain that the company could have when no fraud occurs. We will use this equation in section IV-C to compare all techniques.

$$EE = \frac{EE_{Technique} - EE_{Real}}{EE_{Max} - EE_{Real}} \quad (7)$$

We are not using precision rate to measure how efficient is a technique due to the unbalanced dataset. A random algorithm model would get a very low precision for chargeback, less than 0.5%. This is the reason why we use the EE that is the most relevant factor in our scenario. Using this concept we also avoid the misunderstand of a high precision when classifying all transactions as valid and none as chargeback.

C. Results

Table II summarizes the results for techniques previously described in Section III. The best result between all techniques was the **NN** in March, with 43.66% of EE. Except BN, October is the worst month for all techniques. It is important to emphasize the most important measurement is the Economic Efficiency (EE), which is represented by *Rank*. We inform about precision and recall, which are traditional classification metrics, however in our problem we want to rank transitions according to a fraud score ranking, thus it is not a typical classification problem.

		BN	**LR**	**NN**	**RF**
Oct.	Prec.	7.05	4.10	7.00	10.17
	Rec.	18.93	27.52	9.00	11.47
	Rank.	0.79	1.98	0.36	0.33
	EE	**25.28**	12.03	11.69	8.13
Nov.	Prec.	14.70	8.33	5.00	19.02
	Rec.	32.38	36.67	39.00	27.01
	Rank.	0.73	1.47	2.57	0.47
	EE	29.70	28.73	**33.64**	22.42
Dec.	Prec.	7.40	3.53	5.00	14.17
	Rec.	21.08	30.20	23.00	14.55
	Rank.	1.16	3.49	1.75	0.42
	EE	16.61	10.64	**20.04**	18.02
Jan.	Prec.	8.78	9.70	6.00	13.11
	Rec.	25.56	21.19	21.00	10.60
	Rank.	1.30	0.98	1.30	0.32
	EE	**16.57**	15.54	11.98	9.90
Feb.	Prec.	7.78	6.06	9.00	7.55
	Rec.	42.96	44.62	19.00	18.36
	Rank.	3.10	4.13	1.03	1.12
	EE	**27.40**	25.75	24.03	12.01
Mar.	Prec.	9.93	5.38	6.00	4.24
	Rec.	43.01	49.94	34.00	32.32
	Rank.	2.22	4.76	3.18	3.91
	EE	35.53	35.61	**43.66**	13.48

Table II
COMPARATIVE RESULTS FOR ALL TECHNIQUES ON THE WHOLE DATASET. ABBREVIATIONS: "PREC." IS PRECISION, "REC." IS RECALL, "RANK." IS RANKING

BN has its lowest gain in October with 14.33% of EE, 7.05% of precision at position 0.79% of the ranking. The best result was achieved in March with 35.53% of EE, 9.93% of precision and ranking coverage with 2.22%. The higher precision value was obtained in November with 32.38% of recall. The higher recall rate is in March with 43.01%.

LR has its lowest gain in October with 12.03% of EE with 4.10% of precision and its best EE is 35.61% in March.

NN presents its worst results in October and January with 11.69% and 11.98%, respectively. Its best result is in March with 43.66% of EE and 6% of precision.

RF has the worst EE in October with 8.13%. Its best is 22.42% of EE in November with precision of 19.02% at 0.47% of the ranking.

Figure 4 shows the EE until 8% of the ranking in March. **NN** presents the best performance until 5.80% of the ranking, and after that it drops and stays bellow **BN** curve. The **RF** stays bellow the others until to the end.

These results shows that all the four algorithms can bring

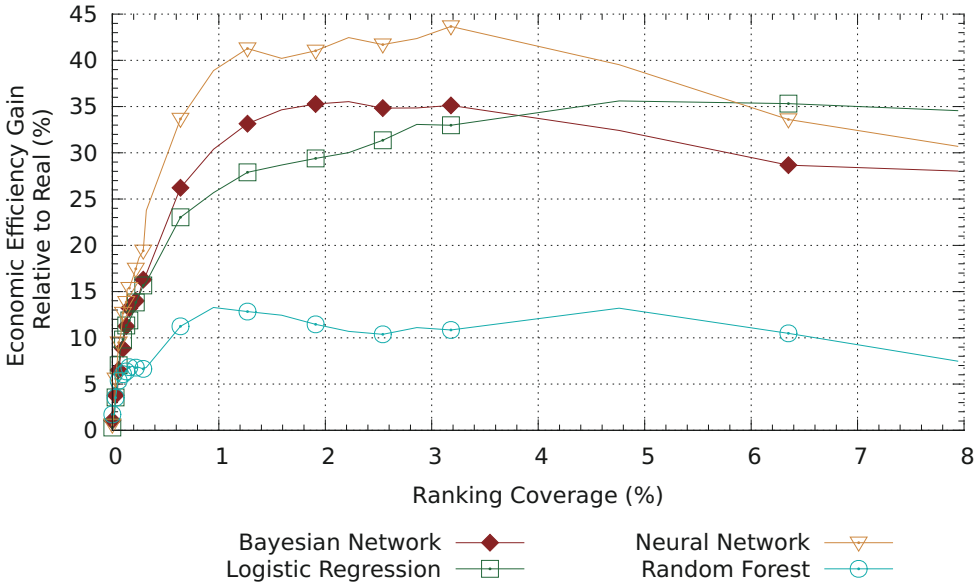


Figure 4. March - EE versus Ranking Position

gains to the company, even the less effective technique reaches at least 8% of Economic Efficiency gain. This methodology of fraud detection can be used by e-commerce companies to reduce the risk in credit card operations. If we compare the techniques to choose the one that would be the best to avoid chargeback, we identify that Bayesian Networks (BN) is the best one, since Neural Networks (NN) presented lower values in some months of the actual dataset. Therefore **BN** has been chosen as the best technique for this scenario, presenting significant gains for all months of data.

V. Conclusion

In this work we build different fraud detection models to predict fraud in online transactions, more specifically credit card operations. We apply and evaluate four different computation intelligence techniques, after choosing them from an initial set of evaluated experiments that adopt several distinct techniques. In order to evaluate the techniques, we apply them in an actual dataset, containing thousands of transactions per day, from the most popular Brazilian electronic payment service, called *PagSeguro*.

We confirm that imbalanced classes, fraud and non-fraud, was a factor that directs impacts on the prediction gains. The achieved results present significant gains when compared to actual scenario of the company, which adopts some fraud detection procedures. In order to compare the techniques, we adopt an Economic Efficiency (EE) function, which describes the financial improvement relative to the actual scenario from the corporation. In the best case, we have achieved a gain of 43.66%.

We realize that the worst results were obtained in the months with a fewer amount of fraud transactions than other ones. Neural Network and Bayesian Networks have performed the best results. The Logistic Regression approach reached its better result in March with 35.61% of EE, slightly better than Bayesian Networks and worst than Neural Networks, with 43.66% of EE. The worst technique was Random Forest with gains in the range of 8.13% to 22.42%.

One of the challenges of this research is the nature of data, since they are much unbalanced with the minor class with less than 1%. As a future work we intend to use techniques to deal with this data imbalance, preserving the generality of the model. One possible solution would be consider weights to assign to classes, where the minor class receives the larger weight [33]. Moreover, Xie et al. [34] have proposed an improvement to Random Forest technique, combining the balanced random forest and the weighted random forest. Thus, an idea is to optimize the computational techniques and another proposal to improve the gains is to use hybrid models that can be composed by ensemble of techniques.

Acknowledgment

This research was supported by the Brazilian National Institute of Science and Technology for the Web (CNPq grant numbers 573871/2008-6 and 477709/2012-5), CAPES, CNPq, Finep, and Fapemig.

References

[1] V. P. Tej Paul Bhatla and A. Dua, *Understanding Credit Card Frauds*, 2003.

[2] C. Mindware Research Group, *2011 Online Fraud Report*, 12th ed., 2011. [Online]. Available: http://www.cybersource.com

[3] S. Bhattacharyya, S. Jha, K. Tharakunnel, Westland, and J. Christopher, "Data mining for credit card fraud: A comparative study," *Decis. Support Syst.*, vol. 50, pp. 602–613, February 2011.

[4] R. J. Bolton and D. J. H, "Statistical fraud detection: A review," p. 2002, 2002.

[5] R. Maranzato, A. Pereira, M. Neubert, and A. P. do Lago, "Fraud detection in reputation systems in e-markets using logistic regression and stepwise optimization," *SIGAPP Appl. Comput. Rev.*, vol. 11, pp. 14–26, June 2010. [Online]. Available: http://doi.acm.org/10.1145/1869687.1869689

[6] P. Ravisankar, V. Ravi, G. R. Rao, and I. Bose, "Detection of financial statement fraud and feature selection using data mining techniques," *Decision Support Systems*, vol. 50, no. 2, pp. 491–500, 2011.

[7] E. W. T. Ngai, Y. Hu, Y. H. Wong, Y. Chen, and X. Sun, "The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature," *Decision Support Systems*, vol. 50, no. 3, pp. 559–569, 2011.

[8] B. Thomas, J. Clergue, A. Schaad, and M. Dacier, "A comparison of conventional and online fraud," in *CRIS'04, 2nd International Conference on Critical Infrastructures, October 25-27, 2004 - Grenoble, France*, 10 2004.

[9] L. Vasiu and I. Vasiu, "Dissecting computer fraud: From definitional issues to a taxonomy," in *Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 7 - Volume 7*, ser. HICSS '04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 70 170.3–. [Online]. Available: http://portal.acm.org/citation.cfm?id=962755.963148

[10] D. H. Chau, S. P, and C. Faloutsos, "Detecting fraudulent personalities in networks of online auctioneers," in *In Proc. ECML/PKDD*, 2006, pp. 103–114.

[11] T. Fawcett and F. Provost, "Adaptive fraud detection. data mining and knowledge discovery," 1997.

[12] E. L. Barse, H. Kvarnström, and E. Jonsson, "Synthesizing test data for fraud detection systems," in *Proceedings of the 19th Annual Computer Security Applications Conference*, ser. ACSAC '03. Washington, DC, USA: IEEE Computer Society, 2003, pp. 384–. [Online]. Available: http://portal.acm.org/citation.cfm?id=956415.956464

[13] C. Phua, V. Lee, K. Smith-Miles, and R. Gayler, "A comprehensive survey of data mining-based fraud detection research," 2005.

[14] S. Maes, K. Tuyls, B. Vanschoenwinkel, and B. Manderick, "Credit card fraud detection using bayesian and neural networks," in *In: Maciunas RJ, editor. Interactive image-guided neurosurgery. American Association Neurological Surgeons*, 1993, pp. 261–270.

[15] Netmap, "Fraud and crime example brochure," 2004.

[16] R. J. Bolton and D. J. Hand, "Unsupervised Profiling Methods for Fraud Detection," *Statistical Science*, vol. 17, no. 3, pp. 235–255, 2002. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.24.5743

[17] G. J. Williams and Z. Huang, "Mining the knowledge mine: The hot spots methodology for mining large real world databases," 1997.

[18] B. Larivière and D. Van den Poel, "Predicting customer retention and profitability by using random forests and regression forests techniques," *Expert Syst. Appl.*, vol. 29, no. 2, pp. 472–484, Aug. 2005. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2005.04.043

[19] A. R. Statnikov, L. Wang, and C. F. Aliferis, "A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification." *BMC Bioinformatics*, vol. 9, 2008. [Online]. Available: http://dblp.uni-trier.de/db/journals/bmcbi/bmcbi9.html#StatnikovWA08

[20] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, May 2011. [Online]. Available: http://doi.acm.org/10.1145/1961189.1961199

[21] E. W. T. Ngai, Y. Hu, Y. H. Wong, Y. Chen, and X. Sun, "The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature," *Decis. Support Syst.*, vol. 50, no. 3, pp. 559–569, Feb. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.dss.2010.08.006

[22] S. Maes, karl Tuyls, B. Vanschoenwinkel, and B. Manderick, "Credit card fraud detection using bayesian and neural networks," *Vrije Universiteir Brussel*, 2001.

[23] G. Cooper and E. Herskovits, "A bayesian method for the induction of probabilistic networks from data," *Machine Learning*, vol. 9, no. 4, pp. 309–347, 1992.

[24] D. W. Hosmer, *Applied Logistic Regression*, 2nd ed. New York: Wiley, 2000.

[25] A. J. Dobson, *An Introduction to Generalized Linear Models*. London:Chapman and Hall, 1990.

[26] W. N. Venables, D. M. Smith, and the R Development Core Team, "An introduction to r," http://www.cran.r-project.org, [Online; Accessed: July 20, 2014].

[27] K. Gurney and K. Gurney, *An introduction to neural networks*. CRC Press, 1997.

[28] A. P. Engelbrecht, *Computational Intelligence: An Introduction*, 2nd ed. Wiley, 2007.

[29] A. Konar, *Computational Intelligence: Principles, Techniques and Applications*. Springer-Verlag New York, 2005.

[30] M. I. A. Lourakis, "A brief description of the levenberg-marquardt algorithm," vol. 3, p. 2, 2005.

[31] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[32] T. R. D. C. T. Version, "R: A language and environment for statistical computing," http://www.r-project.org, [Online; Accessed: July 20, 2014].

[33] C. Chen, A. Liaw, and L. Breiman, "Using random forest to learn imbalanced data," *Discovery*, no. 1999, pp. 1–12, 2004.

[34] Y. Xie, X. Li, E. Ngai, and W. Ying, "Customer churn prediction using improved balanced random forests," *Expert Systems with Applications*, vol. 36, no. 3, pp. 5445–5449, 2009.

Where Should I Go? City Recommendation Based on User Communities

Ruhan Bidart, Adriano C. M. Pereira, Jussara M. Almeida, Anisio Lacerda
Department of Computer Science,
Universidade Federal de Minas Gerais (UFMG) - Brazil
{*ruhanbidart, adrianoc, jussara, anisio*}*@dcc.ufmg.br*

***Abstract*—Recommender systems play a key role in the decision making process of users in Web systems. In tourism, it is widely used to recommend hotels, tourist attractions, accommodations, etc. In this paper, we present a personalized neighborhood-based method to recommend cities. This is a fundamental problem whose solution support other tourism recommendations. Our recommendation approach takes into account information of two different layers, namely, an upper layer composed by cities and a lower layer composed by attractions of each city. It consists of first building a social network among users, where the edges are weighted by the similarity of interests between pairs of users, and then using this network as a component of a collaborative filtering strategy to recommend cities. We evaluate our method using a large dataset collected from TripAdvisor. Our experimental results show that our approach, despite being simple, outperforms the precision achieved by a state-of-the-art baseline approach for implicit feedback (WRMF), which exploits only the overall popularity of cities. We also show that the use of a secondary layer (attraction) contributes to improve the effectiveness of our approach.**

***Keywords*-Social Network; Collaborative Filtering; e-Tourism; Recommendation Systems**

I. Introduction

The explosive growth of content available on the Web has made the process of information search and selection an increasingly complex task. Users are often overwhelmed by the abundance of choice with which they are presented. Recommender Systems (RSs) are information filtering mechanisms devoted to overcome problems that are inherent to information overload [2].

Originally, RSs have been successfully applied on e-commerce web sites to suggest information on items and products that are related to the preferences of the user (e.g. news, web pages, movies, books, etc.). More recently, they have been also applied in the field of electronic tourism (e-tourism), providing services to assist users with travel plans [10]. A travel plan consists of a number of stages, such as: (i) choosing destinations, (ii) selecting tourist attractions, (iii) choosing accommodations, (iv) deciding routes, among others.

In this paper, we are interested in the first stage, which consists in recommending a set of cities to the user. Many travel recommendation systems focus on the second stage – suggesting a set of tourist attractions – assuming that the destination is given [10], [22], [23]. Thus, our present effort complements prior work by focusing on a more fundamental problem, which in turn supports user decisions when building a travel plan.

The key challenge in developing a system for personalized city recommendation is the integration of heterogeneous travel information. The recommendation process should take into account not only a set of different cities which are candidates for recommendation but also the attractions that would lead users to visit each city. Since cities and attractions are different types of evidence used to describe user preferences, it is difficult to automatically combine both of them into a recommendation function.

In this paper, we present a system for personalized city recommendation that takes into account the user preferences for cities and the attractions (e.g., restaurants, movie theaters, etc.) available in each city. The proposed solution is based on *Collaborative Filtering* that relies only on past user behavior (e.g., the cities each user has visited and *liked*) and does not assume explicit profiles [12]. More specifically, we present a neighborhood-based method (NBM) that is centered on computing the relationships among users.

Traditional NBMs compute the similarities between the users/items using the co-preference information, and new items are suggested based on these similarity values [7]. Our proposed method computes the similarity between pairs of users considering the cities previously visited by those users. However, unlike traditional NBMs, it also considers how similar users rated different attractions in each candidate city to select which cities should be recommended to a target user.

Specifically, we start by building a network, in which the nodes are users and the weighted edges represent the similarities among users, considering the visited cities. We then use this network to extract communities of users that have similar interests and preferences. Finally, we recommend cities, considering information about attractions of each city, to a given user using information from the community to which the user belongs.

We evaluate our recommendation method, called ReCWEE – ***Re****commendation using* ***C****ommunities and*

978-1-4799-6953-1/14 $31.00 © 2014 IEEE
DOI 10.1109/LAWeb.2014.15

***W**ithout **E**xplicit **E**valuations* – using a large dataset collected from TripAdvisor, currently one of the most popular travel websites, with nearly 260 million unique monthly visitors[1]. In order to assess the benefits of using information regarding attractions for recommending cities, we consider two variations of ReCWEE: one exploits mainly information regarding cities, whereas the other fully exploits both layers by aggregating information about cities and attractions in each city. We refer to the latter as ReCWEE+. We compare both methods against a state-of-the-art baseline, the Weighted Regularized Matrix Factorization (WRMF) [9]. Our experimental results show that ReCWEE improves the precision of the recommendations in up to 6.8%, on average, which can be attributed mainly to the use of communities. Further improvements (5.2%, on average) can also be obtained when our approach fully exploits both cities and attractions for users with not so sparse profiles.

The rest of this paper is organized as follows. Related work is discussed on Section II. In Section III, we define the problem of personalized city recommendation. We describe the proposed solution in Section IV. Our experimental methodology and main results are presented in Sections V and VI. Section VII offers conclusions and directions for future work.

II. Related Work

The idea behind recommender systems is to automatically recommend items for an user, aiming to predict the user's interest level about the items [21]. These systems help users to deal with information overload, providing personalized recommendations of products and services [2].

Recommender systems are typically classified in three categories: *content-based*, which recommends items that are similar to items that the user preferred in the past [15]; *collaborative*, which recommends items that users with similar profiles have preferred in the past [7]; and *hybrid* approaches, which combine those two to make recommendations [3].

Collaborative Filtering (CF) is one of the most successful recommendation strategies to date [17], and is used in many domains, such as social streams [5] and movies [9]. This approach represents the state-of-the-art among recommendation techniques, and it is used as the basis of our proposed method. There are two primary types of recommendation methods that are based on CF: *neighborhood methods* and *latent factor methods*. We briefly discuss prior work in these two categories next.

A. Neighborhood Methods

Neighborhood methods are centered in computing the similarities among users, and the user/item preferences are directly used to suggest new items [12]. The key problem in these methods is identifying groups of users with similar tastes. There are several techniques to identify communities in graphs [6], the simplest and most generic of them is the k-NN ($k-$Nearest Neighbor), which is used in several works [4], [8].

Formally, the utility $u(c, s)$ of an item s to an user c is based on the utilities $u(c_j, s)$ assimilated to the item s, by users c_j that are "similar" to user c. Although this task is well defined, there are several ways to tackle this problem. In [7], a framework to implement a solution to this problem is defined, which is used as basis for our proposed method. The implementation of this framework is detailed in Section V.

The aforementioned technique can be divided into three main components: (a) similarity computing, (b) neighborhood selection, and (c) top-N recommendation. The literature is rich in various efforts to develop and test algorithms that implement each of these components. For example, in [6], the authors describe and discuss various techniques to compute similarity between different users. The neighborhood selection step was addressed primarily by [7], [11], and more recently by [21], where the authors applied it in a collaborative system using a real social network (Last.FM). Solutions to the top-N recommendation step are specially developed in [8], where several techniques are empirically analyzed. Moreover, in [4], the authors compare evaluation metrics to the top-N recommendation task, which consists in recommending only the best N items that would be suited to the user.

B. Latent Factor Methods

Latent factor methods, such as Singular Value Decomposition (SVD), consist of an alternative approach that maps both items and users into the same latent factor space, thus making them directly comparable [12]. Such methods have become popular by combining good scalability with accurate predictive power.

A representative example of latent factor model is the Weighted Regularized Matrix Factorization (WRMF) [9]. This algorithm is considered the state-of-the-art matrix factorization model when considering implicit feedback. Since in our scenario users do not explicitly state their preference, we use WRMF as a strong baseline for our neighborhood based approach. In WRMF users and items are mapped in a join latent factor space of dimensionality f, such that user-item interactions are modeled as inner products in that space. Each item i is associated with a vector $q_i \in \mathbb{R}^f$ and each user u is associated with a vector $p_u \in \mathbb{R}^f$. The resulting scalar product $q_i^T p_u$, captures the interaction between u and i, leading to the estimation $\widetilde{r}_{ui} = q_i^T p_u$. The challenge here is to compute the mapping of each item and user factor vectors ($q_i, p_u \in \mathbb{R}^f$). With these values calculated we can easily estimate the rating that an user will give to any item.

[1] http://www.tripadvisor.com/PressCenter-c6-About_Us.html

The estimations of q_i and p_u are calculated minimizing the following equation [13]:

$$\min_{q*,p*} \sum_{(u,i)\in K} (r_{ui} - q_i^T p_u)^2 + \lambda(||q_i||^2 + ||p_u||^2) \quad (1)$$

Latent factor models have been successfully applied in different scenarios. For instance, in [18] the authors apply it to the recommendation of products in an e-commerce website, whereas in [14] the authors use existing location data in photos to infer user travel routes and, then, recommend routes to other users. Other prior studies that exploited WRMF are [16], [20], where the authors aim at recommending attractions[2]. Here we use WRMF as a baseline for our neighborhood-based approach, but we focus on recommending cities, while the focus of other works is in the recommendation of attractions. As far as we know, this is the first work that focus on city recommendation by exploring both user preferences for cities and also attractions associated to these preferred cities.

III. Problem Statement

We here propose a collaborative filtering approach to recommend cities to a given user. To that end, we address three main problems.

The first problem is the inference of the weights assigned to links connecting pairs of users. This problem is described as: let U = $\{u_1, u_2,...,u_n\}$ be the set of all users, and $\mathbf{T}$ $\in \mathbb{R}^{nxn}$ a matrix representing similarities between them. Our goal is to fill matrix $\mathbf{T}$, assigning weights to specific pairs of users, which ends up generating a virtual social network between users.

The second problem is to split users into groups (communities). It can be defined as: given an user u (target of recommendation) and a similarity matrix $\mathbf{T}$, the task is to find a group G_u of users that would better describe u (i.e., that are more similar to u). Note that this task is analogous to the task of community detection.

The third problem is a learning to rank task, whose goal is to rank a list of candidate cities for each user. The problem can be defined as follows. Given a target user $u \in U$ that has a group of related users G_u = $\{u_1, u_2,..., u_m\}$, the set C_g = $\{c_1, c_2,..., c_k\}$ of all cities visited by users in G_u, and the set C_u = $\{c_1, c_2,..., c_l\}$ of cities visited by u, our task consists of ranking the set of cities in C_g - C_u based on the usefulness, as a recommendation, of each city to u. Note that we only recommend cities that have not been visited by the target user.

IV. Proposed Solution

The solution we propose to address the city recommendation problem is called ReCWEE – ***Re****commendation using* ***C****ommunities and* ***W****ithout* ***E****xplicit* ***E****valuations*. The hypothesis behind it is that users who visited a large number of cities in common with target user u have a higher probability of visiting other cities that u would like to visit. In other words, the best candidate cities for recommending to u are among those visited by users with similar interests.

The design of ReCWEE assumes that there is no explicit evaluations of cities. This is the case of, for instance, TripAdvisor, which is currently the world's largest travel site according to comScore[3]. It assumes, however, that a set of cities previously visited by each user u, C_u, is given, as is the case of TripAdvisor. The algorithm uses a top-N recommendation approach, taking into account only the group of cities $C_g - C_u$ (see section III).

ReCWEE is composed of three main modules, which work independently and in series, as illustrated in Figure 1. The following sections describe each step of our solution in detail. We finish this section putting them together to build our recommendation strategy.

Figure 1. Overview of our approach. Step 1: a graph of users is created. Step 2: communities in the graph are detected. Step 3: a ranking of cities is generated and the top-k cities are recommended to the user.

A. Graph Generation

The first step in the process of graph generation is related to the question: How can we create a social network linking users? Or, in other words, how can we infer similarities between pairs of users? Since we assume that no explicit user rating about cities is available, we use the set C_u of cities previously visited by user u as evidence of u's interests. Specifically, we estimate the similarity between two users u_1 and u_2 by the similarity between their sets of visited cities, C_{u1} and C_{u2}, computed using the Jaccard coefficient [19] as follows:

$$J(C_{u1}, C_{u2}) = \frac{|C_{u1} \cap C_{u2}|}{|C_{u1} \cup C_{u2}|} \quad (2)$$

Note that we need to calculate the Jaccard coefficient between each pair of users, thus completely filling the matrix $\mathbf{T}$. We did experiment with other similarity metrics (e.g., *cossine similarity*) but the results produced with the Jaccard coefficient are at least as good as those obtained with the other metrics. We thus opted for using Jaccard because, besides the good performance, it is simple and easy to interpret.

[2]In the tourism domain, attractions are locations that can be visited in a city (e.g., museums, movie theaters, etc.).

[3]http://www.comscore.com.

Next, we have to define a threshold τ of minimum similarity. All links with weights below τ will be pruned (removed from **T**), and only the remaining links will be passed to the next step (community detection). Moreover, users that become disconnected from the rest (i.e., isolated nodes in the graph) after the link pruning will also be removed from the graph, since they will not belong to any community.

Rather than arbitrarily selecting a threshold, we opted for first analyzing the distribution of similarity values. To that end, we plotted the number of users that are removed from the graph for various values of threshold. Figure 2 shows this distribution for the dataset we use in this work, which will be presented in the next section.

Note that the distribution has a clear knee: when the threshold τ is set to 0,2, only 3,78% of the users are removed. Moreover, around 95% of the links are removed. We do want a threshold that leads to a minimum number of users removed but a large number of links removed, as those links represent weak connections between users. Thus, the choice of 0,2 seems good. We did experiment with larger values but found no improvements in recommendation precision, experimentally confirming the hypothesis raised in [7] that higher thresholds do not improve recommendation effectiveness.

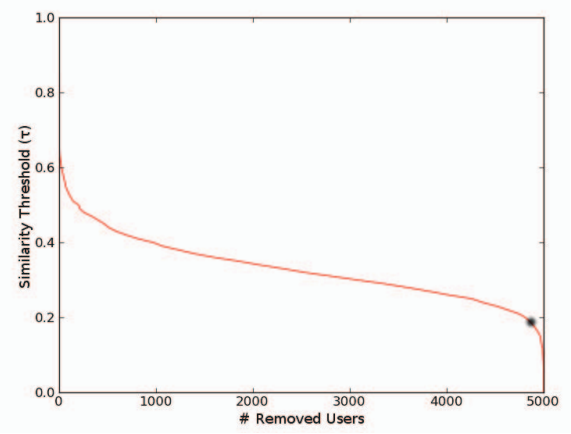


Figure 2. Number of users removed from the graph for each value of threshold. Note that the knee of the curve is around 0,2 (marked with a black circle).

B. Community Detection

In this step we use the k-Nearest Neighbor (k-NN) algorithm to infer the community of a given user u. It simply selects the k users that have the highest similarities with u as its community. Although many other community detection algorithms could be employed in this step [6], [7], we chose k-NN as it is simple and scalable to the volume of data used in this work.

The value of k was set to 20, as suggested by [7], which means that the 20 most similar users to the target user will be selected as its community. Note that user u_2 may belong to the community of u_1, while u_1 may *not* belong to the community of u_2.

The communities generated by k-NN play a key role in the next step of the method (ranking). The set of candidate cities to recommend to a target user u will be extracted from u's community. Thus, by reducing the set of neighbors to the top-20 most similar ones, we are indirectly reducing the search space for cities to be recommended to u.

C. Ranking of Candidate Cities

In this final step, the cities that are candidates to be recommended to target user u are ranked according to the *usefulness* of them to u. The candidate cities are extracted from u's community: they correspond to all cities that were visited by users in u's community (G_u) but not by u, that is, cities in set $C_g - C_u$ (see section III). The usefulness of each candidate to u is estimated by a *score* function. Given this score function, the candidates are simply ranked in decreasing order of score.

One key contribution of this work is propose a score function that uses information of two layers - cities and city attractions. To assess to which extent the use of the lower layer - city attraction - improves recommendation effectiveness, we also consider an alternative score function that exploits only information about the cities.

The simplest function assigns a score to a city c candidate to recommendation to target user u according to the number of users in u's community that had visited c in the past and the average of rankings given to all attractions of this city. That is, this score simply takes the popularity of each city in u's community (G_u), weighting each city also by the overall opinion about its attractions. This score function is defined as:

$$Score_1(c,u) = Popularity(G_u, c) \times MeanAttractionEvaluations(c) \quad (3)$$

where $Popularity(G_u, c)$ is the fraction of users in G_u that had visited c in the past.

We also propose a score function that takes into account not only the popularity of a candidate city in G_u but also how each user in G_u evaluated each of its attractions, that is:

$$Score_2(c,u) = Popularity(G_u, c) \times \sum_{u_i \in G_u} \frac{\frac{1}{|Attr_c|} \sum_{a \in Attr_c}^{n} \frac{Eval(a,u_i)}{Bias(u_i)}}{WithRating(G_u, c)} \quad (4)$$

where $Attr_c$ is the set of attractions in city c, $Eval(a, u_i)$ is the rating that user u_i gave to attraction a in a 1-5 range, and $Bias(u_i)$ is the average of all ratings given by the user u_i. We normalize the evaluations of u_i by this factor so as to better capture any bias u_i might have towards giving higher or lower ratings, and thus more accurately

group the opinions of different users in G_u regarding each attraction. The innermost summation aggregates the opinions of neighbor u_i with respect to all attractions in city c. We then divide this number by the number of attractions in c to take an average opinion of u_i regarding c, and thus avoid favoring cities with a larger number of attractions. Finally, we take the average of these opinions across all neighbors that have rated at least one attraction in the city (outermost summation, where $WithRating(G_u, c)$ gives the number of users in G_u who rated any attraction in city c) and multiply it by the popularity of the city in the group.

We note that ratings of attractions are typically very sparse (as we observed in our dataset). In particular, there might be no ratings about attractions of the city in the community of the target user, or even in the whose dataset. Thus, we propose to carefully adapt how our $Score_2$ function is applied depending on the available data. The main issue regards the set of user G_u in Equation 4. We envision three scenarios:

1) If there are ratings for attractions of the city from the user community: in this case we use the opinions of users in this community, as in Equation 4;
2) If there are no ratings for attractions of the city in the user community, but there are ratings from other users (outside the community): in this case, we replace G_u in Equation 4 by all users in the graph that have rated some attraction in that city.
3) If there are no ratings of attractions from, then our score function reduces to only evaluating the $Popularity$ of the city in G_u.

D. Putting All Together: ReCWEE

The pseudocode of $ReCWEE$ is shown in Algorithm 1. The algorithm receives as input a target user u as well as the set of all users U.

The algorithm starts by generating a graph (line 25). The function $GraphGeneration$ (lines $1-12$) creates a complete graph between users, using Jaccard index of the cities visited by each one of them, and then prunes weak links (and isolated nodes) according to a pre-defined similarity threshold τ.

Next, the k-NN algorithm is used to generate a community for target user u (line 26). Finally, the function Ranking is called to produce a ranking of the candidate cities. The top-N cities are recommended (line 27). Note that the first step, which is more costly, is done offline, at a training phase. Only the community detection and the ranking need to be performed at recommendation time.

The function $Ranking$ (lines $14-21$) gets the target user u and her/his community as parameters, and returns a ranking of cities to be recommended to this user. It first determines the candidate cities by taking cities that users in u's community have visited and removing those that u has already visited, aiming at recommending only cities that the user has not visited yet. Each city receives a score (line 17), using Equation 3 or Equation 4. The list of candidate cities sorted by score in reverse order is then returned (line 19).

Algorithm 1 ReCWEE

```
1: function GraphGeneration(U, τ)
2:     Graph ← ∅
3:     for u1 in U do
4:         for u2 in U do
5:             if u1 ≠ u2 then
6:                 Graph[u1][u2] ← Jaccard(Cities(u1), Cities(u2))
7:             end if
8:         end for
9:     end for
10:    Prune Graph according to threshold τ
11:    return Graph
12: end function
13:
14: function Ranking(u, Gu)
15:    CandidateCities ← Cities(Gu) − Cities(u)
16:    for c in CandidateCities do
17:        c ← score(c, u) ▷ Equation 3 (ReCWEE) or Equation 4 (ReCWEE+)
18:    end for
19:    rank ← SortReverseOrder(CandidateCities, score)
20:    return rank
21: end function
22:
23: function ReCWEE(u, U)
24:    τ ← 0, 2
25:    Graph ← GraphGeneration(U, τ)
26:    Gu ← GenerateCommunitiesKNN(u, Graph)
27:    return top-N in Ranking(u, Gu)
28: end function
```

In order to distinguish between the two proposed score functions (Equations 3 or 4), we refer to our recommendation approach exploiting both cities and attractions (i.e., using Equation 4) as ReCWEE+. It is important to note that, though ReCWEE+ is explained based on the *cities* x *attractions* scenario, it can be easily generalized to other scenarios that have hierarchical structure. For example a system where user gives ratings to answers and you want to recommend categories of answers (*categories* x *answers* scenario).

V. Experimental Methodology

In this section, we first present our dataset and the data acquisition process (Section V-A). We then present the methods used as baselines in our evaluation (Section V-B), and briefly describe our evaluation setup (Section V-C).

A. Data Acquisition

The dataset used in this work was collected from TripAdvisor [1], and covers a period of over 5 months, from October 23^{rd} 2013 to March 23^{rd} 2014. We focus on recommending cities to Brazilians. Thus, as a starter to our crawling process, we select two groups of seeds: one containing the 30 most popular touristic cities in Brazil, and the other with the 15 most popular touristic cities in the world. These rankings of most popular cities are taken from TripAdvisor's Web site.

Starting with the set of seeds, our crawler proceeds as follows: it first collects the city page and then gathers

each attraction found in it. For each attraction, it collects all reviews posted by users regarding that attraction. After that, it downloads the profile of the review's author, which includes the list of cities previously visited by this user. Finally, it repeats the process for each new city discovered, as illustrated in Figure 3.

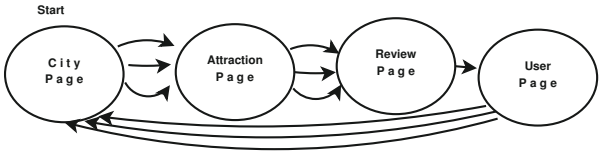


Figure 3. Overview of our TripAdvisor crawler. It starts with a set of cities (seeds), visits all attractions that are listed on it. After that, it collects all reviews of each attraction. For each review it collects the user that wrote the review. Finally, it get all cities visited by the user and follows it, restarting the process. Transition with one arrow means that the *from* page has only one entity that leads to the *to* page, while in transitions with more than one arrow, the page *from* leads to several entities *to*.

Table I summarizes our dataset, presenting the numbers of the main entities – pages, cities, attractions, reviews and users – available. We here focus on three of those entities, namely *Cities*, *Attractions* and *Users*. We leave the design of recommendation strategies that also exploit user reviews as a future work. It is important to emphasize that all users were anonymized for privacy reasons. For each user, our dataset contains an identifier, a location, a list of cities she/he has visited (city *seenlist*), a list of attractions she/he has rated (attraction *seenlist*). Each attraction is associated with a name, an address, a city, an average user rating, and a number of reviews.

Table I
OVERVIEW OF OUR DATASET COLLECTED FROM TRIPADVISOR

Entity	Number
Pages	1,597,609
Cities	85,505
Attractions	162,168
Reviews	599,629
Users	266,392

B. Baseline Methods

We compare our proposed methods against the following two baselines:

1) **Popularity**: this is a very basic and intuitive strategy that always recommends the most popular cities, regardless of the target user. In other words, it is a non-personalized recommendation strategy. The popularity of a city is estimated by the number of users who have previously visited it, i.e., the number of users who have that particular city in their city seenlists. This strategy, though naive, has shown to be surprisingly powerful in various applications [9];
2) **Weighted Regularized Matrix Factorization (WRMF)**: this algorithm is considered to be the state-of-the-art for implicit feedback recommendation, therefore it is a strong baseline and difficult to outperform. In our case, we modeled WRMF with users and cities (items). WRMF receives the relations between users and cities (implicit), discovers latent factors about them and, after that, uses these latent factors to recommend cities to the users (for more details about WRMF see Section II-B).

C. Evaluation Setup

In our study, we use a fully automatic evaluation methodology, as most previous work [4], [7]–[9], [14]. Specifically, we adopt a five-fold cross-validation procedure. That is, the dataset is randomly split into 5 pieces (folds), each one containing 20% of the *cities* and associated information (attractions and users). Four folds are used for training the recommendation methods, i.e., for computing the similarities between users as well as the popularity of cities, and one fold is used as test set to evaluate the methods. The folds are switched and the process is repeated 5 times, each one with different training and test sets. We note that, in the particular case of ReCWEE and ReCWEE+, only data in the training set is used in the graph generation phase. Thus, only cities in the training set can be considered candidates for recommendation.

The test sets are used to evaluate the recommendations. A city is considered a relevant recommendation to a target user u if it appears in the city seenlist of u (that is, we do have evidence that u visited the city) and is in the *test* set[4], that is, from the perspective of the recommendation strategy, it is unknown.

We use **precision** and **recall** in the top-k recommended cities as main evaluation metrics. Let C_u be the sorted list of recommended cities produced by the method being evaluated to a target user u, and C_u^k the top k cities in this list. Let also Rel_u be the set of relevant cities for user u (extracted from the test set). The precision in the top-k recommendations, p_k, is defined as:

$$p@k(C_u^k, Rel_u) = \frac{|C_u^k \cap Rel_u|}{min(k, |C_u^k|)} \quad (5)$$

The recall in the top-k recommendations is defined as:

$$recall@k(C_u^k, Rel_u) = \frac{|C_u^k \cap Rel_u|}{min(k, |Rel_u|)} \quad (6)$$

All tests were executed five times with a different selection of users in each of them. It is reported the confidence interval (95%) of these results. Following, we present the results.

[4]Otherwise, if it is in u's seenlist, it must be in the training set and thus is not considered as a candidate for recommendation.

VI. Results

This section presents results of the experiments, applying our two proposed algorithms (ReCWEE and ReCWEE+) in a real dataset collected from TripAdvisor, and comparing these results with two baselines. At first, we are not worried about comparing performance of the algorithms, because all the process must be executed offline and only once a day. An important point to be emphasized about our experiments is related to the number of cities on users' seenlists. As expected, the distribution of the seenlists length by number of users follows a long tail distribution, as shown in Figure 4, indicating that most users have seenlists with only a few cities.

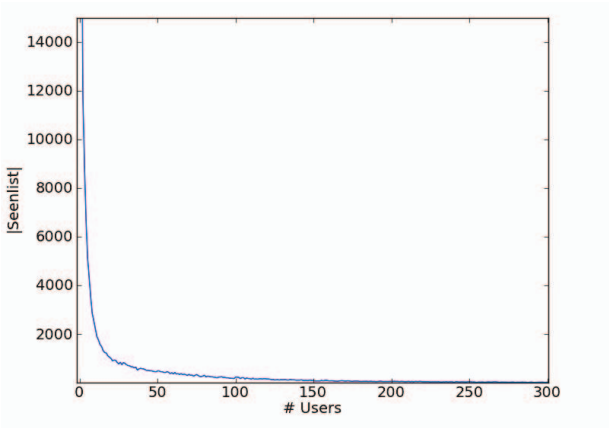


Figure 4. Distribution of seenlist's length by users.

Thus, two types of results will be presented, one for users with small seenlists and another for users with big seenlists. Section VI-A presents the experiments for a larger group of users (small seenlists) and section VI-B corresponds to an analysis of a group of users that have big seenlists.

A. Results for groups with small seenlists

As the focus of this work is not the *cold-start* problem, it was chosen to filter users that have seenlists with less than 10 cities, which resulted in 63,822 users. Nevertheless, these users are from several nationalities and it is desired to work only with Brazilians, because the objective of this work is to understand their behavior and recommend cities to them. Thus, the data was filtered to select only Brazilians, but because of unstructured data and by the lack of location data for several users, this filter resulted in 26,549 users.

It was generated rankings for the four algorithms (Popularity, ReCWEE, ReCWEE+ and WRMF) in five groups of 5,000 users, chosen randomly from this 26,549, and these groups can share users between them. For each one of these five groups, we executed tests with five-fold cross-validation. The results to these experiments are reported in Figure 5, with confidence interval of 95%. In the case of WRMF we have tried several different parameters configuration, the best of them was $\lambda = 150$ and $number\ of\ factors = 100$ (as suggested in [9]). This configuration set was used in all experiments that are shown in this section.

It is important to note that in Figure 5 the algorithms ReCWEE and ReCWEE+ are significantly better than the other two in terms of precision. We also can see that the difference between our two proposed approaches was small. In the next section we will show that it is because of the lack of data to improve recommendations. Once the sole difference between ReCWEE+ and ReCWEE is the use of attractions, when these data is very much sparse or there is no data, ReCWEE+ can not outperforms ReCWEE. Note that in neither case ReCWEE overcomes ReCWEE+, as expected.

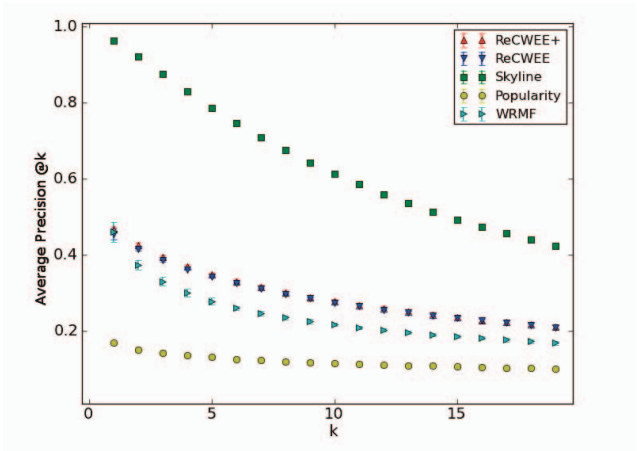


Figure 5. Precision of the four algorithms and *Skyline* (maximum possible precision for ReCWEE and ReCWEE+). All results are reported with confidence interval of 95%. Note that the algorithm ReCWEE and ReCWEE+ are significantly better than the other two, an that the difference between our proposals, with this amount of available data, is small. It occurs due to missing data about attractions and even because of small seenlists.

In Figure 5 we also report the *Skyline*. These green squares are a measure of the maximum that our approaches could reach. It is because we use a neighborhood method and it filters data by similar users before recommending. Green squares show the maximum that could be achieved using groups selected by our neighborhood selection approach. They also show that our two first steps (see section IV), which filters candidate cities, are well done, once it is observed a great distance between *Skyline* and all methods.

Recall results are presented in Figure 6. It can be seen that ReCWEE and ReCWEE+ algorithms have better recall than the other two baseline algorithms. Green squares present the maximum recall that could be reached using our neighborhood model. Note that it shows again that our method of filtering candidate cities is good because all methods are far

away from the upper limit (*Skyline*).

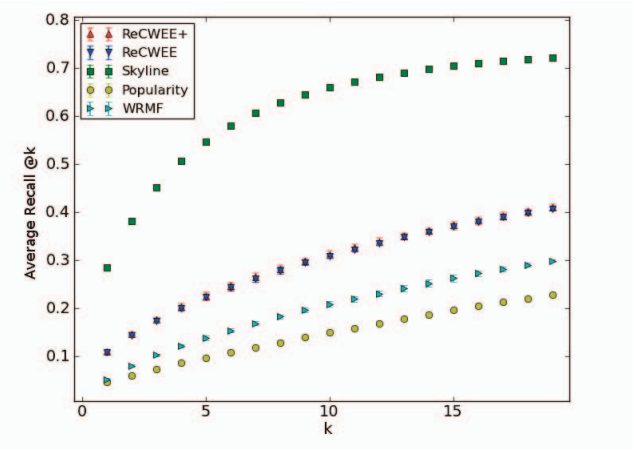


Figure 6. Recall results to the four algorithms and the *Skyline* limit for our proposed methods. All results are presented with confidence interval of 95%. Note that ReCWEE and ReCWEE+ are also significantly better than the others in terms of recall but not significantly different to each other.

B. Results for groups with big seenlists

The same tests were executed for $|seenlist| \geq 100$, aiming to check the hypothesis that ReCWEE+ should achieve better precision for users with more cities in their seenlists. The results of these experiments are presented in Figure 7.

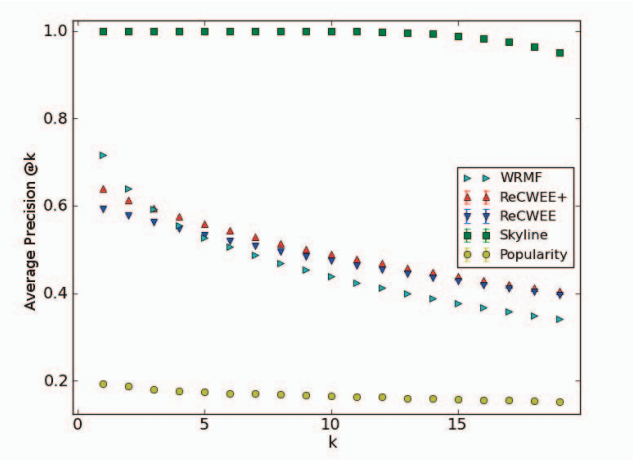


Figure 7. Precision for groups of users with big seenlists.

The most important aspect to note in Figure 7 is that with more data ReCWEE+ outperforms ReCWEE in 4.2% when considering Precision@5 (on average), confirming our hypothesis that the use of a second layer information (attractions) can improve our recommendation method.

Another important aspect is that for users with more data in their profiles WRMF outperforms ReCWEE and ReCWEE+ on the first two positions of the ranking. Although it is an advantage for WRMF, it is important to realize that it overcomes our methods only in the two first ranking positions. In all next ranking positions our proposed solution outperforms the baseline, which is very important in our case because recommending cities makes more sense when you recommend at least five cities, providing more choices to the user. Despite seeing this behavior of WRMF, in recall (see Figure 8) ReCWEE+ is always better than WRMF, in all ranking positions.

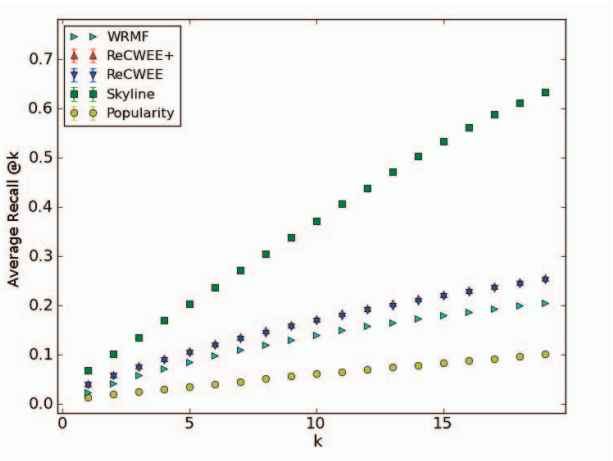


Figure 8. Recall for groups of users with big seenlists. Note that recall for ReCWEE and ReCWEE+ is better than WRMF.

The result analysis also allows to understand that it is possible to create groups containing only users with more data in their seenlist, in order to develop a hierarchical recommendation task, where groups of users with similar number of cities in their seenlists would be segmented and used together to get a better precision using only their relations (reduction in sparsity by removing users with small seenlists).

VII. Conclusion

Recommendation is a way of helping users in the increasingly difficult task of making decisions on-line, whatever the nature of the decision. In this work, the task of recommending cities was faced for the first time, as far as we know. This task is shown as being important for the simple fact that there are systems, such as TripAdvisor, where people share their opinions about the most different destinations with the goal of helping each other to make decisions about traveling.

First, we formalize the problems addressed in this paper. Then, we detail a methodology to solve each one of these problems. The proposed methodology, although it has been applied in an actual case study, is not specific and can be

used in different scenarios, establishing itself an important contribution of this work.

Our results showed that infer a social network on TripAdvisor, along with detecting communities on it, is a good technique to improve precision in recommendation of cities that users have not visited yet.

The results also show that the use of a secondary data layer (attractions) brings benefits to our method, increasing substantially precision in comparison with the method that does not use a lower layer. ReCWEE+ also outperforms the state-of-the-art method (WRMF) for users with a small data volume and, for those with big volume of data, ReCWEE+ overcomes WRMF after the position @3 in the ranking.

Finally, the findings show that techniques of segmentation and hierarchy of users could be used to further improvements in precision, since the split of users with a higher information showed a higher precision.

As future work it is intended to improve the currently ranking technique, taking into account the diversity of recommendations. It is also intended to use data from reviews of each attraction in order to find out more descriptive characteristics of cities, making it possible to group cities by its similarities, which would enable to recommend cities by groups not only individually. Additionally, it is intended to link users by a real social network, using data from $Facebook$ that are available for a representative set of users of our TripAdvisor's dataset.

ACKNOWLEDGMENT

This research was supported by the Brazilian National Institute of Science and Technology for the Web (CNPq grant numbers 573871/2008-6 and 477709/2012-5), CAPES, CNPq, Finep, and Fapemig.

REFERENCES

[1] "Tripadvisor," http://tripadvisor.com.br, 2014.

[2] G. Adomavicius and A. Tuzhilin.

[3] M. Balabanović and Y. Shoham, "Fab: Content-based, collaborative recommendation," *Commun. ACM*, vol. 40, no. 3, pp. 66–72, Mar. 1997.

[4] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-n recommendation tasks," in *Proceedings of the 4th ACM Conference on Recommender Systems*, 2010, pp. 39–46.

[5] E. Diaz-Aviles, L. Drumond, L. Schmidt-Thieme, and W. Nejdl, "Real-time top-n recommendation in social streams," in *Proceedings of the Sixth ACM Conference on Recommender Systems*, ser. RecSys '12, 2012, pp. 59–66.

[6] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3-5, pp. 75 – 174, 2010.

[7] J. Herlocker, J. A. Konstan, and J. Riedl, "An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms," *Inf. Retr.*, vol. 5, no. 4, pp. 287–310, Oct. 2002.

[8] J. L. Herlocker, J. A. Konstan, L. G. Terveen, John, and T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems*, vol. 22, pp. 5–53, 2004.

[9] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *Proceedings of the Eighth IEEE International Conference on Data Mining*, ser. ICDM '08, 2008, pp. 263–272.

[10] Y. Huang and L. Bian, "A bayesian network and analytic hierarchy process based personalized recommendations for tourist attractions over the internet," *Expert Systems with Applications*, vol. 36, no. 1, pp. 933–943, 2009.

[11] M. Jamali and M. Ester, "Using a trust network to improve top-n recommendation," in *Proceedings of the 3rd ACM Conference on Recommender Systems*, 2009, pp. 181–188.

[12] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 426–434.

[13] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[14] T. Kurashima, T. Iwata, G. Irie, and K. Fujimura, "Travel route recommendation using geotags in photo sharing sites," in *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, ser. CIKM '10, 2010, pp. 579–588.

[15] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, Jan. 2003.

[16] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo, "A random walk around the city: New venue recommendation in location-based social networks." in *SocialCom/PASSAT*. IEEE, 2012, pp. 144–153.

[17] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds., *Recommender Systems Handbook*. Springer, 2011.

[18] J. B. Schafer, J. A. Konstan, and J. Riedl, "E-commerce recommendation applications," *Data Min. Knowl. Discov.*, vol. 5, no. 1-2, pp. 115–153, Jan. 2001.

[19] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Addison Wesley, 2006.

[20] H. Wang, M. Terrovitis, and N. Mamoulis, "Location recommendation in location-based social networks using user check-in data," in *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2013, pp. 374–383.

[21] X. Yang, H. Steck, Y. Guo, and Y. Liu, "On top-k recommendation using social networks," in *Proceedings of the Sixth ACM Conference on Recommender Systems*, ser. RecSys '12, 2012, pp. 67–74.

[22] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee, "Exploiting geographical influence for collaborative point-of-interest recommendation," in *Proceedings of the 34th international ACM SIGIR Conference on Research and development in Information Retrieval*, 2011, pp. 325–334.

[23] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann, "Time-aware point-of-interest recommendation," in *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2013, pp. 363–372.

A personalized geographic-based diffusion model for location recommendations in LBSN

Iury Nunes
Federal University of Campina Grande
Campina Grande, Brazil
iury.nunes@ccc.ufcg.edu.br

Leandro Marinho
Federal University of Campina Grande
Campina Grande, Brazil
lbmarinho@dsc.ufcg.edu.br

***Abstract*—Location Based Social Networks (LBSN) have emerged with the purpose of allowing users to share their visited locations with their friends. Foursquare, for instance, is a popular LBSN where users endorse and share tips about visited locations. In order to improve the experience of LBSN users, simple recommender services, typically based on geographical proximity, are usually provided. The state-of-the-art location recommenders in LBSN are based on linear combinations of collaborative filtering, geo and social-aware recommenders, which implies fine tuning and running three (or more) separate algorithms for each recommendation request. In this paper, we present a new location recommender that integrates collaborative filtering and geographic information into one single diffusion-based recommendation model. The idea is to learn a personalized ranking of locations for a target user considering the locations visited by similar users, the distances between visited and non visited locations and the regions he prefers to visit. We conduct experiments on real data from two different LBSN, namely, Gowalla and Foursquare, and show that our approach outperforms the state-of-art in most of the cities evaluated.**

***Keywords*-Recommender Systems, Location Based Social Networks, Collaborative Filtering, Diffusion Model, Location-Aware**

I. Introduction

Location-based Social Networks (LBSN) or Geosocial Networks are online social networks in which geographic services and capabilities are used to let users check-in and then broadcast their locations. Prominent examples of LBSN are Foursquare[1] and Facebook places[2] whose main goal is to help people meet up with friends and discover interesting places. Thanks to the affordable prices of GPS enabled smartphones and other mobile devices, these services are now widely spread and poised for continued growth. Being released in 2009, Foursquare, for example, reached over 50 million users and 6 billion check-ins worldwide in 2014[3]. When the space of choices available (which places to visit?) gets too large, the problem known as information overload emerges.

[1]http://foursquare.com
[2]http://www.facebook.com/about/location
[3]https://foursquare.com/about

Recommender systems appear as a natural solution to this problem offering effective techniques for helping users to filter out and discover relevant information in large data sets. Foursquare uses the current location of users to recommend nearby locations based on existing categories. For example, users might want to know which restaurants he can visit nearby his current location. Although this certainly helps users to reduce the space of choices available, it does not take into account their personal preferences. Users might be interested, for example, in nearby restaurants within a certain price range or that offer live music. In fact, the state-of-the-art works on location or points-of-interest (POI) recommendations propose a combination of collaborative filtering (for capturing personal POIs preferences of users) and some recommender that captures the preferences of users for geographic regions (aka geographic-aware recommender). In many cases, a social-aware recommender is also included in the ensemble [3, 17, 18]. Although these algorithms have presented promising results, they require fine tuning and execution of two or more specialized algorithms for each recommendation request. Moreover, the personal preferences of users for POIs and geographic regions are modeled separately, while there is a clear correlation between them, e.g., the preference of a user for a region may reinforce the preferences of this user for the POIs in that region and vice-versa.

In this work, we propose to model the preferences of users for locations, the distances between visited and non visited locations, and the preferences of users for geographic regions in a weighted graph. We then use a random walk algorithm for capturing, in a transparent way, the interplay between these three kinds of user preferences. We conduct experiments on real data from two different LBSN, namely, Gowalla and Foursquare, and show that our approach outperforms the state-of-art in most of the cities evaluated. Our contributions are as follows:

- We propose a new model for check-in data that is able to capture, at the same structure, the distances between locations and the preferences of users for POIs and geographic regions.

978-1-4799-6953-1/14 $31.00 © 2014 IEEE
DOI 10.1109/LAWeb.2014.22

- We propose to exploit the interplay between these three kinds of data through a diffusion model, i.e., a random walk algorithm.
- We evaluate our approach on two large-scale data sets and show that our approach outperforms the state-of-the-art algorithms in most of the cities evaluated.

The rest of the paper is organized as follows. In Section II we formalize the problem approached by this paper. In Section III we present the related work and position this paper among them. In section IV we recall the basic concepts of the random walk technique. In Section V we describe our approach in detail. In Section VI we present the evaluation protocol used and discuss the outcome of the experiments. Finally, Section VII concludes the paper and discusses the outlook.

II. Problem Setting

The recommendation scenario that we investigate in this paper is as follows: a user specifies the city of interest and the recommender engine suggests locations (or POIs) within the selected city that are likely to be relevant to the user. It is important to remark that the city of interest might be either the user's hometown or a city where the user is traveling. The only requirement is that the user must have at least one check-in in the city of interest.

Thus, let U be the set of users, L the set of locations and C the set of cities. In this paper we only consider implicit feedback data, i.e., the set $S \subseteq U \times L \times C$ of ternary relations between users and geotagged venues. The task is then to find a scoring function

$$\hat{s} : U \times L \times C \rightarrow \mathbb{R} \tag{1}$$

that assigns a preference score for locations within a certain city, given a target user. Thus, for a given user $u \in U$, the topN recommendations can be computed by

$$\mathit{topN}(u, c) := \operatorname*{argmax}^{n}_{l \in L_c \setminus L_u} \hat{s}(u, l, c) \tag{2}$$

where n denotes the number of locations to be recommended, L_c the set of locations within city c and L_u denotes the set of locations checked-in by user u. For convenience, we also define U_l as the set of users that checked-in at l.

III. Related Work

Several research works have exploited check-in data of LBSN as a useful source of information for understanding human mobility patterns [5, 4, 12, 16]. Although these works are not directly related to recommender systems, the insights they provide can be used to devise novel and effective location-aware recommendations. For example, two important insights coming from some of these works are: (i) the distribution of distances between pairs of visited locations by a user resembles a powerlaw, i.e., the majority of locations that a given user has checked-in at are close to each other; and (ii) users tend to concentrate their check-ins around a few regions, e.g., their homes and/or workplaces.

This topic of research is closely related to context-aware recommendations, in which the context of interest is the geographic position of items and/or users. The geographic context is very challenging by itself [13], but other contexts were also investigated by the literature, like the timestamp [6] and the category of checked-in locations (e.g., "food", "museum", "stadium", etc.) [1, 11]. Although these contexts are indeed important and worth investigating, in this work we focus on the the geographic context given that this is the core feature of LBSN.

Most of the existing research work on location recommendation proposes some sort of combination between collaborative filtering and some recommender that exploits the geographic preferences of users. Cheng et al. [3] combined, through a simple multiplication, a probabilistic factor model for collaborative filtering with a Multi-center Gaussian Model for modelling the geographic preferences of the users. Ye et al. [17] in turn, fused, through a simple linear combination, neighborhood-based collaborative filtering with a power law based model for modelling the geographic preferences of users. We present more details about these two recommenders in Section VI-B. Instead of using different recommendation algorithms in an ensemble as the aforementioned works do, we define a single recommendation algorithm that takes into account both collaborative filtering and the geographic preferences of users in a transparent fashion.

More recently, location recommenders based on probabilistic topic models have appeared [7, 10, 18]. Differently from us, the works of [7, 10] do not use data from LBSN, but from services like yelp, twitter and flickr. Although there are geotagged data in these services, they do not convey the same information that check-in data does. For example, the fact that a given geotagged photo was shared in flickr does not imply that the user who uploaded the photo wanted to share the location where the photo was shot. Differently from these two works, [18] used data from LBSN and EBSN (Event Based Social Networks) to evaluate their recommendation model. Their approach proves to outperform the method of Ye et al. [17] in several scenarios, except in the scenario we investigate in this paper, i.e., recommending for a user who has some chek-ins in the city he is located and is looking for recommendations within that city. Hence, we will compare our approach with the approaches of [17] and [3].

Random walk-based recommendation models have proved to be very effective in domains other than location recommendation [8, 2, 9]. We complement these works introducing a new information diffusion model that captures, at the same time, collaborative filtering, the distance between POIs and the geographic preferences of users.

IV. Random Walk on Graphs

Before we introduce our diffusion-based recommendation model in the next section, we briefly recall the basic principles of information propagation on graphs using random walk. A popular implementation of random walk is the PageRank algorithm, which exploits the hyperlink structure of web pages under the assumption that a web page is important if there are many pages linking to it, and if those pages are important themselves [14].

A graph is defined as a tuple $G = (V, E)$ where V is a set of vertices (or nodes) and E a set of edges $E \subseteq V \times V$. An edge $e_{u,v} \in E$ means that there is a link between $u \in V$ and $v \in V$. A graph can be weighted in order to denote that some edges are more important than others. Thus, let $w : E \to \mathbb{R}$ be the function that assigns weights to edges in G.

The idea behind random walk algorithms is that there is a "walker" visiting vertices randomly in the graph. At each iteration, the "walker" will jump from one node to another according to the weights of the edges leaving the current node. The larger the weight of an edge, the larger the probability of the "walker" passing through that edge. As the number of iterations increases, the "walker" will visit some nodes more often than others. The idea is that the nodes that were visited more frequently should be ranked higher because they are more important.

We can also represent a graph G as an adjacency matrix $A \in \mathbb{R}^{|V|\times|V|}$ where the value at entry $A_{i,j}$ corresponds to $w(e_{i,j})$. If we model the weights of each edge $e_{i,j} \in E$ as the probability of the "walker" jumping from the node i to node j in the next iteration, then we can compute the probabilities of the user being at any node given a number of iterations t. As the number of iterations grows, these probabilities will converge to a steady state. So, let $\vec{x}$ be a vector with $|V|$ dimensions. At each iteration, this vector will be updated with a probability distribution of the nodes.

Since the probability distribution of $\vec{x}$ will eventually converge, it doesn't really matter in which node the walker starts. So, at iteration $t = 0$ we randomly choose any vertex $v \in V$ and assign the value 1 to it, i.e., $\vec{x}_0[v] = 1$ and 0 for all the other nodes $u \in \{V \setminus \{v\}\}$. Now, at each subsequent iteration we update the values of $\vec{x}$ as described in Equation 3 where $\vec{p}$ is a vector (typically filled with a uniform probability distribution) that can be used for asserting preferences for specific nodes and the teleport factor $0 < \lambda < 1$ is used for determining the strength of the influence of $\vec{p}$. In practice, the teleport factor refers to the probability of the "walker" jumping to any other node in the graph, even the nodes that are not linked to the current node. The teleport factor is important to increase the probability of visiting vertices that have only a few or no incoming edges. The process stops when there is no significant change between $\vec{x}_t$ and $\vec{x}_{t+1}$.

$$\vec{x}_{t+1} = \lambda A^T \vec{x}_t + (1 - \lambda)\vec{p} \tag{3}$$

Other versions of this algorithm also uses the concept of restart. While the teleport allows the "walker" to jump uniformly to any node, the restart factor will bias the jump towards the starting node. As we will see in Section V-A, we modeled each user as a node. Since the goal of the recommender is to generate personalized recommendations for a given target user, the node of such user should be considered more important than the other nodes. So, if we set the target user node as the starting node and increase the influence of the restart factor, we guarantee that the "walker" will visit this node more often. Thus, the steady state achieved by the random walk will be biased towards the target user personal preferences.

V. A Diffusion Model for Location Recommendations

Inspired by the findings of related works, we designed a location recommender that takes the following assumptions into account:

1) Users that visited similar locations in the past tend to visit the similar locations in the future [17].
2) Users tend to visit locations close to the locations they have already visited in the past [4, 12, 13].
3) Users tend to concentrate their check-ins around a few regions of interest [5, 3].

In subsection V-A we present a graph representation for check-in data. Next, in subsections V-B, V-C and V-D we describe, step by step, how we can approach each one of these three assumptions using a graph-based diffusion model. Finally, in subsection V-E these three assumptions will be integrated into a single recommendation model.

A. Modelling Check-in Data on a Graph

We structure the implicit feedback data of LBSN as a graph where nodes are comprised of users and locations, assuming that these users and locations belong to some given city. There is an edge between a node $u \in U$ and a node $l \in L$ if user u has checked-in at l. If we define a Boolean function checkedIn(u, l) to denote whether the user u checked-in at location l, we can formally define graph G as follows: $G = (V, E)$ where $V = U \cup L$ and $E = \{\{(u, l) : u \in U, l \in L, \text{checkedIn}(u, l) = \mathbf{true}\} \cup \{(l, u) : u \in U, l \in L, \text{checkedIn}(u, l) = \mathbf{true}\}\}$.

Figure 1 depicts an example of this graph where $U = \{u_1, u_2, u_3\}$ and $L = \{l_1, l_2, l_3, l_4, l_5\}$. Since one of the main goals of recommender systems is to help users finding new items (in our case locations) we will not consider the locations that the target user $u_1 \in U$ has already checked-in at. The locations l_1 and l_3 are colored in red in Figure 1 to denote that these locations have been already checked-in by

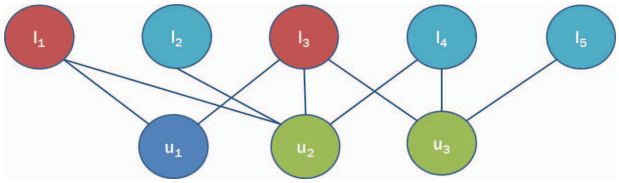


Figure 1: Graph depicting the relation between users and their checked-in locations.

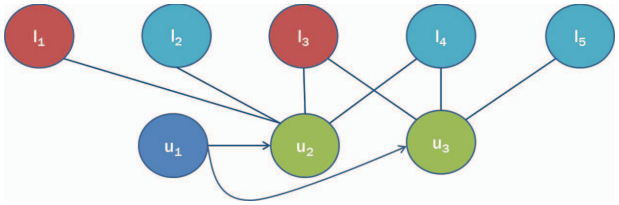


Figure 2: Collaborative Filtering Graph

the target user. Thus, the only recommendable locations for user u_1 are l_2, l_4 and l_5.

If we weigh each outgoing edge, for any node in V, with the same probability and perform a random walk over this weighted graph restricting the final probabilities to the nodes of type location, we would end up with a ranking that corresponds to the most popular locations, i.e., the locations that were checked-in more often by distinct users. More formally, let $\text{outDegree}(u) = |\{v \in V : (u,v) \in E\}|$ be the out degree of a given node $u \in V$ of the graph, then the weights of the graph take the form of Equation 4.

$$w(u,v) = \frac{1}{\text{outDegree}(u)} \tag{4}$$

Notice that the popularity-based recommender does not provide personalized recommendations so we will not consider it in our model. In the following subsection we will describe how to achieve personalization by means of collaborative filtering.

B. Diffusion-based Collaborative Filtering

The collaborative filtering assumption is that users who shared the same interests in the past tend to share the same interests in the future. The user-based collaborative filtering (based on K-nearest neighbors) is a classic recommender that, despite its simplicity, has proven to attain high accuracy in LBSN [17].

To implement this algorithm we need to define a measure that captures the similarity between a pair of users. We have chosen to use the well known cosine similarity measure which have been successfully used in many domains including LBSN [17]. First, we define, for each user $u \in U$, a vector $\vec{u}^{|L|}$ whose components are equal to 1 if the user checked-in at the corresponding location and 0 otherwise. The cosine similarity between any two users $u, v \in U$ is then defined as follows:

$$sim^{cf}(u,v) = \frac{\vec{u} \cdot \vec{v}}{||\vec{u}||||\vec{v}||} \tag{5}$$

Now, for finding the k-nearest neighbors of a given target user $u \in U$ we only need to compute the similarity between u and every other user and sort these users in descending order of similarity down to k.

Let N_u be the set of k-nearest neighbors of target user u and

$$L_{N_u} = \bigcup_{v \in N_u} L_v$$

be the set of all locations visited by all neighbors of u. Now, a user-based collaborative filtering graph can be defined as $G_{cf} = (V_{cf}, E_{cf})$ where $V_{cf} = \{u\} \cup N_u \cup L_{N_u}$, $E_{cf} = \{(u,v) : v \in N_u\} \cup \{(v,l) : v \in N_u, l \in L_{N_u}\} \cup \{(l,v) : l \in L_{N_u}, v \in N_u\}$ and the edge weights are defined as:

$$w^{cf}(p,q) = \begin{cases} \frac{sim^{cf}(p,q)}{\sum_{v \in N_u} sim^{cf}(p,v)}, & \text{if } p = u \text{ and } q \in N_u \\ \frac{1}{|L_p|}, & \text{if } p \in N_u \text{ and } q \in L_{N_u} \\ \frac{1}{\text{outDegree}(p)}, & \text{if } p \in L_{N_u} \text{ and } q \in N_u \\ 0, & \text{otherwise} \end{cases}$$

Notice that the values of the weights of the edges are normalized so that the sum of the weights of the outgoing edges of each node is not greater than 1. Figure 2 depicts how the graph of Figure 1 becomes a collaborative filtering graph assuming that $u_2 \in U$ and $u_3 \in U$ are the target user neighbors. In this example, the weights of each outgoing edge of u_2 would be 0.25 and the weights of the incoming edges of u_2 from l_1, l_2, l_3 and l_4 would be 1, 1, 0.5 and 0.5 respectively. Notice that if we apply a random walk on this graph, location l_4 would be ranked higher than l_2 and l_5 because l_4 can be reached through u_2 and u_3 whereas l_2 can only be reached through u_2 and l_5 can only be reached through u_3.

C. Diffusion-based Pairwise Distances

It is not a surprise that the check-ins of the users are not uniformly distributed over the map. It generally demands time and money to visit locations which are far from users homes. Thus, people tend to visit locations that are close to the locations they already visited in the past [4, 12].

We capture this assumption in the graph G_{dist}, where edges between locations are created and weighted with the distances (in kilometers) between them. As a similarity measure between two locations l and l', we use the one defined in Equation 6, where $\text{dist}(l,l')$ computes the geographical distance between locations l and l'.

$$sim^{\text{dist}}(l,l') = \min(1, \frac{1}{\text{dist}(l,l')}) \tag{6}$$

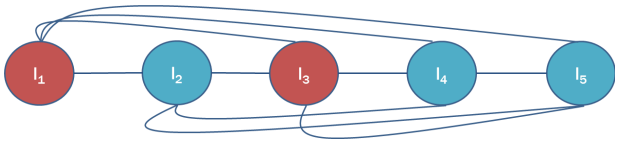


Figure 3: Distance Based Model Example

We use the function min, instead of $1/\text{dist}(l, l')$, in order to have a similarity measure in the range $[0, 1]$. If we used only the inverse of the distance as the similarity between these locations, we would have a similarity equal to 2 if the distance between the locations was equal to 0.5 kilometers, for example.

Now we can formally define G_{dist} as follows: $G_{\text{dist}} = (V_{\text{dist}}, E_{\text{dist}})$ with $V_{\text{dist}} = L$, $E_{\text{dist}} = L \times L$ with the edge weights defined in Equation 7. Notice that the weights of the edges are simply the similarity measure defined in Equation 6 normalized.

$$w^{\text{dist}}(p, q) = \frac{sim^{\text{dist}}(p, q)}{\sum_{l \in V_{\text{dist}} \setminus \{p\}} sim^{\text{dist}}(p, l)} \tag{7}$$

Figure 3 depicts the locations presented in the previous figures but now with links between them. If we now perform a random walk on this graph assuming that all the locations are geographically disposed as presented in Figure 3 and that $\text{dist}(l_1, l_2) = \text{dist}(l_2, l_3) = \text{dist}(l_3, l_4) = \text{dist}(l_4, l_5)$, we would rank l_2 and l_4 on the top (both having the same ranking score) followed by l_5.

D. Diffusion-based Regions of Interest

Users tend to check-in locations in a few well defined regions [5, 3]. For example, two typical regions where users tend to concentrate their check-ins are the regions around their homes and work.

In order to model this assumption we first need to infer the regions of interest for users since this is not given in the check-in data set. For that, we employed the same approach as [5], where the world is discretized into a grid so that each cell becomes a region. In our case, we considered 20 by 20km cells. Although in [5] it was used 25 by 25km cells, we achieved better results decreasing the size of the cell to 20km.

Let R be the set of regions, L_r the set of locations within region $r \in R$, and $L_{u,r}$ the set of checked-in locations user u has done at region r. Now, for a given target user $u \in U$ we define a graph for modelling user preferences for regions as $G_{\text{reg}} = (V_{\text{reg}}, E_{\text{reg}})$ where $V_{\text{reg}} = \{u\} \cup R \cup L$, $E_{\text{reg}} = \{(u, r) : r \in R\} \cup \{(r, l) : r \in R, l \in L_r\}$ and the edge weights are defined as follows:

$$w^{\text{reg}}(p, q) = \begin{cases} \frac{|L_{p,q}|}{|L_p|}, & \text{if } p = u \text{ and } q \in R \\ \frac{1}{|L_p|}, & \text{if } p \in R \text{ and } q \in L_p \\ 0, & \text{otherwise} \end{cases}$$

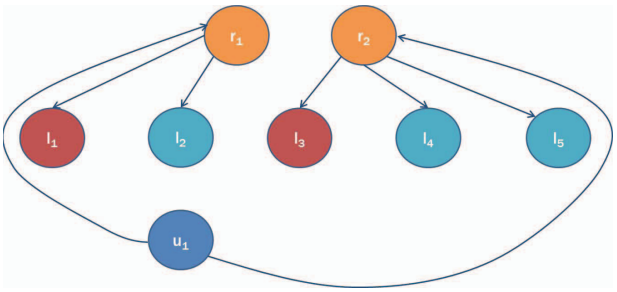


Figure 4: User Region Preferences

For each target user we consider only his regions of interest, disregarding the regions of interest of other users. Figure 4 depicts the graph of Figure 1 with the regions of interest of the target user included. In this example, the target user has equal preferences for regions r_1 and r_2 with one check-in at each region, although there is only one possible location to be recommended in region r_1, which is l_2, whereas there are two locations to be recommended in r_2. When we apply random walk on this graph, both regions will be visited with the same approximate frequency by the random "walker", although he will have less options within r_1 than r_2. Thus, l_2 will probably be more visited than l_4 and l_5. For this reason, l_2 would be ranked higher than l_4 and l_5 in this model.

E. Putting Everything Together

In this subsection we are going to show how to combine the three models presented in the previous subsections into one unified diffusion model. For a given target user $u \in U$, let $G_{\text{unif}} = (V_{\text{unif}}, E_{\text{unif}})$ where $V_{\text{unif}} = V_{cf} \cup V_{\text{dist}} \cup V_{\text{reg}}$, $E_{\text{unif}} = E_{cf} \cup E_{\text{dist}} \cup E_{\text{reg}}$ with edge weights defined as:

$$w^{\text{unif}}(p, q) = \begin{cases} \alpha w^{cf}(p, q), & \text{if } p = u \text{ and } q \in N_u \\ \beta w^{cf}(p, q), & \text{if } p \in N_u \text{ and } q \in L_{N_u} \\ \gamma w^{cf}(p, q), & \text{if } p \in L_{N_u} \text{ and } q \in N_u \\ \delta w^{\text{dist}}(p, q), & \text{if } \{p, q\} \subseteq L_{N_u} \cup L_u \\ \theta w^{reg}(p, q), & \text{if } p = u \text{ and } q \in R \\ w^{\text{reg}}(p, q), & \text{if } p \in R \text{ and } q \in L_p \\ 0, & \text{otherwise} \end{cases}$$

It is important to remark that the values of the hyperparameters $\alpha, \beta, \gamma, \delta$ and θ cannot be greater than 1. Moreover, since α and θ weight edges leaving the target user, $\alpha + \theta$ must not be greater than 1. Similarly, γ and δ weigh edges leaving locations, thus $\gamma + \delta$ must not be greater than 1 either.

Differently from the works of [3, 17], we are not generating two or more ranking scores for being combined, but we are rather using one single model that generates one single ranking taking all the assumptions presented in the beginning of this section into account.

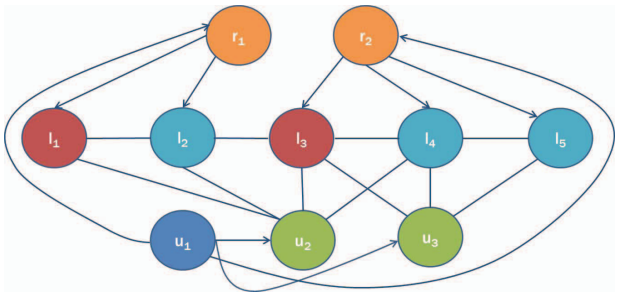


Figure 5: Unified Recommender Model Example

Figure 5 depicts the graph representing our final model. We omitted some edges between the locations is order to improve the readability. For generating recommendations, we perform a random walk on this graph. After the algorithm finishes, we sort the locations in descending order of probability and select the top-5 locations. We will refer to this model as Diffusion Geographic Model (DGM).

VI. Evaluation

In this section we describe the experimental protocol, data sets, compared recommenders and results achieved. We consider the recommendation scenario where we know exactly the city where the user is currently located. This is a realistic scenario with practical applications since the information of the current city of a user is easy to access in GPS enabled devices, like smartphones and tablets. Moreover, it is more likely that users will visit locations within the city he is currently located than traveling to visit a location in other (sometimes distant) cities.

A. Data Preparation

We conducted experiments on data from Foursquare[4] and Gowalla[5], two LBSN data sets that were extensively used in the related works [4, 5]. As is usual in the recommender systems literature, we focused on the dense part of the data, i.e., we are not considering the check-ins of the users that have visited less than 10 locations as well as the locations that were visited by less than 10 users.

Before running the recommendation algorithms, we partitioned the original data sets into cities and selected the top-5 cities in number of check-ins. We ended up with three cities from Foursquare (New York, Los Angeles and Chicago) and two from Gowalla (Austin and Stockholm). The characteristics of each city, after filtering the locations and users, are summarized in Table I.

[4]The data was collected by the authors of [4] from September 2010 to January 2011 and is available under request at: http://infolab.tamu.edu/data/

[5]Gowalla was bought by Facebook on 2011 and discontinued on 2012. The data set we used was collected by the authors of [5] from Feb. 2009 to Oct. 2010 and is available at: http://snap.stanford.edu/data/loc-gowalla.html

City	# Check-ins	$\|U\|$	$\|L\|$	Sparsity	LBSN
New York	184,760	12,005	3,073	99.49%	Foursquare
Austin	127,625	5,385	4,000	99.40%	Gowalla
Stockholm	90,851	5,559	3,850	99.57%	Gowalla
Los Angeles	64,494	6,317	1,274	99.19%	Foursquare
Chicago	54,600	5,268	1,090	99.04%	Foursquare

Table I: Data Set Statistics

B. Compared Algorithms

We compared our approach against the two state-of-art location recommender models described at [3] and [17]. Both approaches use a combination of collaborative filtering, a geographic-aware recommender that models the geographic preferences of users and a social-aware recommender. In this work, we will user only the collaborative filtering and the geographic-aware components of each approach. There are two main reasons that lead us to ignore the social-aware component. The first one is that the information of friendship relations was not collected in the foursquare dataset we used in this work. The second reason is that according to Ye et al. [17] the social relation influence the check-ins mostly when the users travel to distant cities, which is not the scenario we are exploring in this work. We briefly recall these two approaches below.

Based on the idea that there are some regions that the user might be more interested than others, Cheng et al. [3] proposed an approach for combining the preferences of users for regions with his preferences for checked-in locations. For that, first the locations checked-in by users are clustered into regions. Each region is assumed to follow a Gaussian distribution, and the relevance of a new location is computed as the weighted sum of the distances between this location and each of the centroids representing the regions of interest of the target user. These distances are weighted by the importance of a region (number of check-ins at the region) and the probability of the location given a region. For collaborative filtering, it was used probabilistic matrix factorization [15] taking into account the frequency of check-ins of users. It is worth to remark that the combination of these two models was done by a simple multiplication. We will refer to this approach as FMFMGM: Fused Matrix Factorization framework with the Multi-center Gaussian Model. The hyperparameter values of the geographic model of FMFMGM we used in our experiments were $d = 15$, $\alpha = 0.2$, $\theta = 0.02$, as defined in [3]. The hyperparameters values of the matrix factorization model of FMFMGM we used in our experiments were $\alpha = 20$, $\beta = 0.2$, $\lambda = 0.001$ and $k = 10$.

Ye et al. [17] showed that the distances between pairs of checked-in locations of a user follow a power law distribution. After applying a logarithm transformation to this data, the authors propose to learn the parameters of a power law distribution using simple linear regression. The relevance of a new location is then a product over the probabilities

(coming from the fitted power law distribution) of the distances between the new location and all the locations already checked-in at by the target user. For collaborative filtering the authors used K-nearest neighbors with the cosine as similarity function. These two models were fused by a simple linear combination. In our experiments, we set the weight of the geographic recommender to 0.05 and the weight of the user K-nearest neighbors recommender to 0.95, cross-validation tuning. Similarly to the original paper will will refer to this model as UG to denote that it combines users (U) preferences and with geographic (G) influences.

C. Evaluation Metrics and Protocol

We split the data sets into two distinct sets, the training and the testing sets. For each user in each city data set we randomly removed 10% of his checked-in locations for testing and used the remaining 90% for training. This process was repeated 10 times for each city in order to avoid taking conclusions from biased data. We have a hit for a target user each time the recommendation list of this user contains a test location.

As evaluation metrics we used precision@5 and recall@5. Let T_u be the set of test locations for a given user $u \in U$ and R_u the top-5 recommendation list for this same user. Then precision@5 and recall@5 for a given target user $u \in U$ are defined as follows:

$$\text{precision@5}(u) = \frac{|T_u \cap R_u|}{|R_u|}, \quad \text{recall@5}(u) = \frac{|T_u \cap R_u|}{|T_u|}$$

The hyperparameter values we used in our model are: $k = 80$ (the number of nearest neighbors), $\alpha = 0.5$, $\beta = 0.25$, $\gamma = 0.9$, $\delta = 0.1$, $\theta = 0.25$, $\lambda = 0.1$ (teleport factor) and the restart probability = 0.01. These values were defined by cross-validation.

D. Results and Discussion

The results of our experiments are summarized in Figures 6 and 7. These figures depict the average recall@5 and precision@5 for each of the cities described in subsection VI-A averaged over all the 10 random training/testing splits.

Notice that our approach outperforms the compared algorithms in most of the cities evaluated in both precision@5 and recall@5. While DGM and UG achieved similar performance, FMFMGM performed very poorly. One of the reasons for this performance is the extreme level of sparsity of the data sets (cf. Table I), making it very difficult for matrix factorization to learn a reasonable model. The related works have shown that in LBSN the geographical data carries, although important, little signal in comparison to collaborative filtering. Thus, if the collaborative filtering component does not work well, the geographic model alone will perform poorly.

City	N. York	Austin	Stockholm	Los Angeles	Chicago
Prec@5	0.122	$1.45E$-05	0.358	0.0006	0.008
Recall@5	0.130	$5.00E$-06	0.0305	0.02019	0.006

Table II: Student's paired t test - UG vs DGM

Although the difference is small in comparison to UG in some cities, the difference is statistically significant. For verifying that we conducted a student's paired t-test considering each city and each metric. The null hypothesis of the test was: the UG performance is equal or better than the performance of our approach. Thus the alternative hypothesis is: the UG performance is worse than the performance of our approach. The p-values of the t-tests are summarized in Table II. Since we wanted to achieve 95% of confidence on our conclusions, if the p-value is less than $0.05(1-95\%)$ we can reject the null hypothesis and conclude that the performance of our approach is better than the performance of the UG approach, otherwise we can conclude that either both performances are equal or that the UG performance is better than the performance of our approach.

As we can see from Table II, our approach is better than the UG approach for almost all cities in all metrics, with 95% of confidence. The null hypothesis was not rejected in only 3 out of the 10 tests, i.e., the precision@5 and recall@5 at New York and precision@5 at Stockholm. Since the null hypothesis was accepted for these 3 tests, we know that for each of these scenarios the UG performance might be better or equal to the performance of our approach. Then, for these three scenarios we conducted a second student's paired t-test. Now, the null hypothesis is: the performances of UG and DGM are equal. The p-values of the precision@5 in New York, recall@5 in New York and precision@5 in Stockholm were, respectively, 0.2453, 0.2604 and 0.7177.

Thus, according to these results, we can affirm with 95% of confidence that our approach performs better than UG in check-in data from Los Angeles, Chicago and Austin or presents the same performance in New York. In Stockholm our approach performed better than UG in recall@5 and presented the same performance in precision@5.

VII. Conclusion

In this work, we introduced a novel diffusion-based location recommendation model, which captures in a single model the users personal taste, the geographic distances between visited locations and regions of interests of the users. This is different from related works where the recommendation models are ensembles of two or three specialized recommenders. We evaluated our model using real world data from two popular location-based social networks: Gowalla and Foursquare. Our experiments showed that our approach outperforms the compared state-of-the-art algorithms in most of the cities evaluated. For future work, we pretend to investigate other contexts of LBSN domain, such as the time of check-ins and the categories of locations.

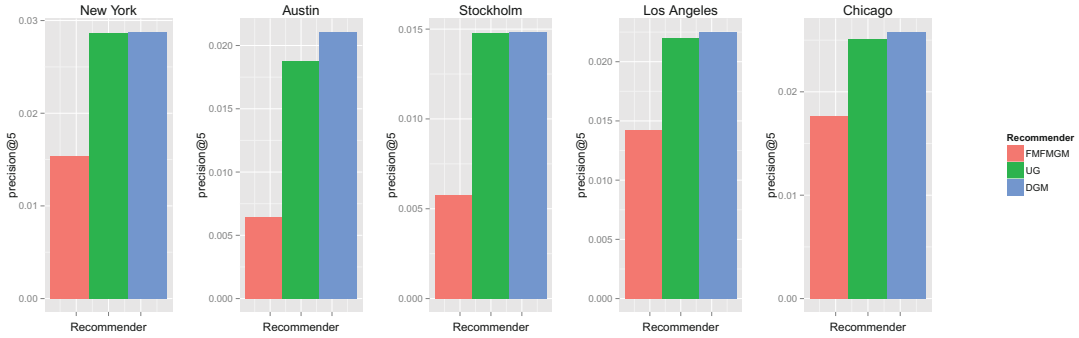


Figure 6: precision@5

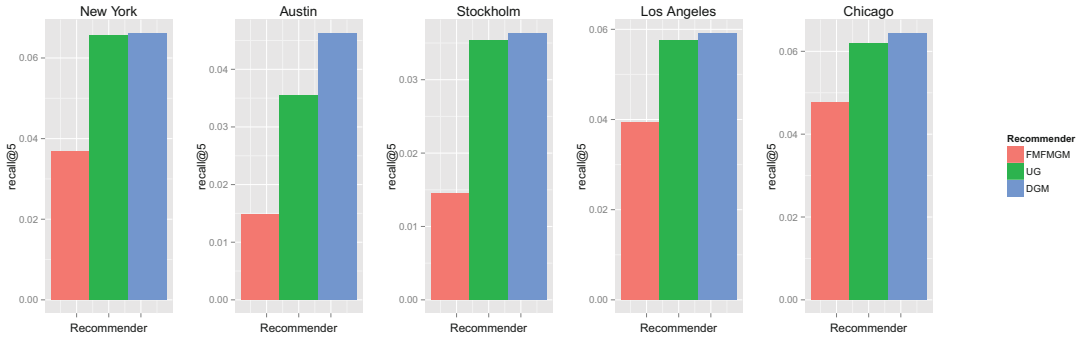


Figure 7: recall@5

References

[1] Jie Bao 0003, Yu Zheng, and Mohamed F. Mokbel. "Location-based and preference-aware recommendation using sparse geo-social networking data." In: *SIGSPATIAL/GIS*. Ed. by Isabel F. Cruz et al. ACM, 2012, pp. 199–208. ISBN: 978-1-4503-1691-0.

[2] Lars Backstrom and Jure Leskovec. "Supervised Random Walks: Predicting and Recommending Links in Social Networks". In: *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*. WSDM '11. Hong Kong, China: ACM, 2011, pp. 635–644. ISBN: 978-1-4503-0493-1.

[3] Chen Cheng et al. "Fused Matrix Factorization with Geographical and Social Influence in Location-Based Social Networks". In: *AAAI*. 2012.

[4] Zhiyuan Cheng et al. "Exploring Millions of Footprints in Location Sharing Services." In: *ICWSM*. Ed. by Lada A. Adamic, Ricardo A. Baeza-Yates, and Scott Counts. The AAAI Press, 2011.

[5] Eunjoon Cho, Seth A Myers, and Jure Leskovec. "Friendship and mobility: user movement in location-based social networks". In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2011, pp. 1082–1090.

[6] Huiji Gao et al. "Exploring temporal effects for location recommendation on location-based social networks." In: *RecSys*. Ed. by Qiang Yang 0001 et al. ACM, 2013, pp. 93–100. ISBN: 978-1-4503-2409-0.

[7] Bo Hu and Martin Ester. "Spatial topic modeling in online social media for location recommendation." In: *RecSys*. Ed. by Qiang Yang 0001 et al. ACM, 2013, pp. 25–32. ISBN: 978-1-4503-2409-0.

[8] Robert Jäschke et al. "Tag Recommendations in Social Bookmarking Systems". In: *AI Commun.* 21.4 (Dec. 2008), pp. 231–247. ISSN: 0921-7126.

[9] Ioannis Konstas, Vassilios Stathopoulos, and Joemon M. Jose. "On Social Networks and Collaborative Recommendation". In: *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '09. Boston, MA, USA: ACM, 2009, pp. 195–202. ISBN: 978-1-60558-483-6.

[10] Takeshi Kurashima et al. "Geo Topic Model: Joint Modeling of User's Activity Area and Interests for Location Recommendation". In: *Proceedings of the Sixth ACM International Conference on Web Search*

and Data Mining. WSDM '13. Rome, Italy: ACM, 2013, pp. 375–384. ISBN: 978-1-4503-1869-3.

[11] Xin Liu et al. "Personalized point-of-interest recommendation by mining users' preference transition." In: *CIKM*. Ed. by Qi He et al. ACM, 2013, pp. 733–738. ISBN: 978-1-4503-2263-8.

[12] Anastasios Noulas et al. "An Empirical Study of Geographic User Activity Patterns in Foursquare." In: *ICWSM*. Ed. by Lada A. Adamic, Ricardo A. Baeza-Yates, and Scott Counts. The AAAI Press, 2011.

[13] Iury Nunes and Leandro Marinho. "A Gaussian Kernel Approach for Location Recommendations". In: *Proceedings of KDMiLe - Symposium on Knowledge Discovery, Mining and Learning, ISSN 2318-1060*. 2013.

[14] L. Page et al. "The PageRank citation ranking: Bringing order to the Web". In: *Proceedings of the 7th International World Wide Web Conference*. Brisbane, Australia, 1998, pp. 161–172.

[15] Ruslan Salakhutdinov and Andriy Mnih. "Probabilistic Matrix Factorization". In: *Advances in Neural Information Processing Systems*. Vol. 20. 2008.

[16] Thiago H. Silva et al. "You are What you Eat (and Drink): Identifying Cultural Boundaries by Analyzing Food & Drink Habits in Foursquare". In: *CoRR* abs/1404.1009 (2014).

[17] Mao Ye et al. "Exploiting geographical influence for collaborative point-of-interest recommendation." In: *SIGIR*. Ed. by Wei-Ying Ma et al. ACM, 2011, pp. 325–334. ISBN: 978-1-4503-0757-4.

[18] Hongzhi Yin et al. "LCARS: a location-content-aware recommender system." In: *KDD*. Ed. by Inderjit S. Dhillon et al. ACM, 2013, pp. 221–229. ISBN: 978-1-4503-2174-7.

A Network Analysis on Movie Producing Teams and their Success

Wladston Viana, Pedro Onofre Santos, Ana Paula Couto da Silva, Mirella M. Moro
Computer Science Departament
Universidade Federal de Minas Gerais, Belo Horizonte, Brazil
Email: {wladston,pedroo,ana.coutosilva,mirella}@dcc.ufmg.br

***Abstract*—We perform social network analysis on movie producing teams formed by directors, producers and writers, using data from IMDb. We assemble an evolving social network by linking agents that worked together throughout history. After, we proceed to calculate topological and non-topological metrics from this network and its teams through time. We present the evolution of topological and non-topological metrics. We analyze the correlation between these metrics and two success parameters: movie's ratings and gross income.**

***Keywords*-Social Networks, Topological and Non-Topological Metrics, Team Success, IMDb.**

I. Introduction

A wide range of human interactions and relationships can be represented through graphs. Besides the well-known online social networks such as Twitter[1] and Facebook[2], social network analysis techniques are also being applied to study collaboration among actors, athletes, executives, musicians, scientists, and many other work environments involving teamwork and group-oriented activity [1], [5], [8], [9], [13], [14].

It is on the best interest of managers and policy makers to form teams in such way that maximizes the productivity. The director of a soccer club would benefit from composing a team with higher winning odds; a faculty director would want to fund research teams that will produce better articles; a manager from a company wants to re-arrange his team in order to ramp up productivity. To help in these important tasks, a huge amount of data regarding how groups work and interact became available in the latest years. With this data, new research revealing previously unknown correlations between properties from the underlying interaction network and the overall success and output quality from its agents are being proposed.

In the entertainment industry, agents team up and work together in order to produce movies, television shows, music albums, Broadway musicals, and many more. Among all entertainment branches, the context in this work is on the film industry. The movie industry per se is a billion dollar business; hence, a movie's public acclaim and critic review play a very important economic role. Using data from the Internet Movie Database (IMDb[3]) to analyze metrics based on team composition and arrangement in the network, we might discover factors associated with the production of better movies. The IMDb is one of the most thorough and detailed cinema database over the internet. An analysis of such extensive data yields more robust and reliable conclusions than many previously conducted experiments performed over smaller data [14].

Note that accessing the relation between topological and non-topological properties of a collaborative network and its success parameters has a high relevance for any industry. Specifically, such relation may guide strategies for organizing teams in a way that optimizes their revenue capacity and social impact. In this work, we study known topological metrics (such as the small world coefficient, betweenness, closeness and local clustering coefficient) applied over the IMDb data for the movie industry. Some metrics are global and relative to the network as a whole, whereas others are local and specific to agents in a single movie producing team. We also study some non-topological metrics, such as past individual experience. We then correlate these metrics with movie's success parameters (rating and gross income).

Next, we discuss the related work (Section II) and the dataset that we analyze (Section III). Then, we go over our main contributions, which are summarized as follows:

- We describe our network model for movie-producing teams composed by producers, directors and writers. We also define topological and non-topological metrics for studying the impact of team composition in the movie success (Section IV).
- We experimentally analyze the correlation between topological and non-topological metrics with movies' rating and gross income success (Section V).

II. Related Work

Understanding how people work together in order to better achieve goals has been explored in many different contexts [2], [3], [4], [5], [6], [8], [9], [11], [12], [13], [14]. Many research papers focus on team formation among scientists and their publication rate and impact factor metrics. For instance, scientific collaboration networks and their properties have been studied by Newman [8], [9]. The author shows that different scientific communities form small-world networks and are highly clustered, and proposes a method for estimating tie strength. Borner et al.[13] explores the

[1]Twitter: http://www.twitter.com
[2]Facebook: http://www.facebook.com
[3]IMDb: http://www.imdb.com

978-1-4799-6953-1/14 $31.00 © 2014 IEEE
DOI 10.1109/LAWeb.2014.10

"Science of team science", a research area focusing on the processes by which scientific teams organize and conduct their work. Such research explores how teams connect and collaborate in order to achieve breakthroughs that would not be attainable by either individual or simple additive efforts.

People also aspire to understand factors that may explain high productivity and success across many scenarios, making network studies of team formation go beyond collaboration among scientists. For instance, Nemoto et al.[6] showed that Wikipedia[4] editors with more social capital (taking part in a cohesive and centralized cluster) produce higher quality articles faster. Singh et al.[12] found that specific kinds of network ties among open source developers are correlated with the development of more popular open source projects.

Other authors explore the network topology of the agents as a tool for understanding their success. Most of them study the correlation between success and the small world coefficient of the network. Chen et al.[2] studied the network formed by collaboration among countries and showed that the small world coefficient is correlated to patent registrations. Schilling and Phelps [11] studied the collaboration among companies and found that the small world metric is correlated to knowledge creation inside companies.

Regarding the entertainment segment, the work by Uzzi and Spiro [14] is the most related to ours. The authors studied the network formed by Broadway musical producers (choreographers, writers and directors, not the cast), and found evidence that the artistic and financial success of such a network as a whole is correlated to its small world coefficient. The authors analyzed many network metrics and found that some of those were correlated to success, while others were not.

To the best of our knowledge, the present work is the first to study the relation between network aspects and success considering motion pictures producers. Furthermore, the dataset is large and composed by several movie genres.

III. Dataset Description

In this work, we analyze the IMDb database, which contains information from thousands of movies from the late 1800's until 2013, from all over the world. For each movie, its list of directors, writers and producers is available, as well as the rating received from IMDb's users. For some movies, the gross income is also available. It is important to state that only movies produced for cinema were analyzed, leaving all TV productions out of the experiment: TV productions are essentially organized differently than cinema productions, and it is debatable if ratings from TV series and movies can be compared to cinema ratings. In total, over 190 thousand of cinema titles were available from the database at the time it was fetched, containing over 320 thousand production team members (directors, writers and producers).

[4]Wikipedia: http://www.wikipedia.org

Most of the movies in IMDb are from extremely unknown productions, which received very little or no user ratings and reviews. Ratings for those movies cannot be compared to well established cinema productions, therefore we decided to filter out those that received less than 25 thousand user votes. That is also a prerequisite for inclusion in the *IMDb TOP250 list* [5], and it clearly selects movies with substantial social impact. We compared this subset of the database with the whole, and it still maintains a similar histogram of number of productions per year, user votes per year and number of agents per team. Also, the non-significant movies only add noise to the correlation analysis, i.e., the dataset without them provide more homogeneous sample. The final subset contains about 1.5% movies (3006 titles) of the total [6].

Evaluating metrics on a network with very few nodes and edges may produce distorted results. It is then necessary to bootstrap the movie producing graph until it reaches a minimum size, i.e., before network metrics become significant. For this reason, we use all movie data from before 1945 just to bootstrap the network with edges and vertices. The experimental analysis considers the whole historical network, but the network metrics and movie success parameters were only extracted for movies produced after 1945.

For evaluating the movie's economic success, we chose the gross income, as it is directly connected with the title's financial revenue and represents how many people were interested in paying to watch such a movie. Also, as a public's acceptance metric, we considered the IMDb user rating, as it indicates how well the title was received by the public. Using these two variables, we are also able to correlate the movie's economic success with its public acceptance.

Ratings for the movies were normalized for the number of votes received using a true Bayesian estimate, which is the same used by IMDb in its TOP 250 movie list:

$$WeightedRating = (\frac{v}{v+m}) \times R + (\frac{m}{v+m}) \times C, \quad (1)$$

where, for each movie, R is the mean of its ratings, v is its number of votes received, m is the least possible amount of votes (25 thousand), and C is the mean vote across the whole report. The value of C is provided by IMDb and it is equal to 7.0. For the TOP 250, only votes from regular voters are considered.

The gross income information used in our work is also present in the IMDb database, but only for a few movies. The gross income value is usually given in the currency of the country that hosted the movie production, and is dated from shortly after the movie's release. In order to accurately compare gross income from different movies with minimal

[5]IMDB TOP250 list: http://www.imdb.com/chart/top

[6]We have also evaluated the results considering the (significantly larger) set of movies (more than thousand ratings), comprising about 9% of the total, and the results were similar.

distortion, the values had to be normalized. Monetary figures for gross income were converted to US Dollars using the Historical Currency Converter Web Service[7]. The corresponding amount in US Dollars was subsequently corrected for inflation considering the present time using the CPI Inflation Calculator[8], an online feature provided by the Bureau of Labor Statistics. Gross income figures not listed in US Dollars that also did not possess a valid historical exchange record, were discarded, as they represented only about 0.2% (6 movies) of the chosen sample.

IV. Network Modeling and Topological Metrics

The IMDb database provides the full cast and crew from movies, including actors, producers, directors, writers, art direction, special effects team, soundtracks and sound effects department, and many more. For modeling a movie-producing team, we decided to include only the producers, directors and writers, leaving out the rest of the production crew and cast. This choice was made because such selected agents are the core of the team: they take the important decisions and hire the rest of the crew. The responsibility for the success of the movie ultimately falls on those agents. The total of agents in our network is $11,832$.

We model the IMDb movie database as a bipartite graph, with edges between a set a movies and a set of selected agents (producers, writers and directors), indicating individuals who produced each movie. Most network metrics in the literature cannot be applied to bipartite networks, so in order to calculate them we projected the network into a one-mode graph. In this projection only agents are present as nodes, and edges exist between agents who worked on a same movie, following the methodology proposed by Newman [8].

Since we are interested in studying the network's evolution through time, we process the dataset in chronological order of movie production. For each movie, we take its producers, writers and directors as vertices, and create unweighted edges between them to indicate existing previous work.

To increase the fidelity of our model to how movie producing parties actually interact, when a node ceases to participate in any movie for more than 7 years, we remove it and all its vertices from the database. We note that such an agent is likely to be retired and thus not participating actively in the network, following the same methodology proposed in [14].

In our analysis, we consider the small world coefficient for measuring the overall cohesion in the entire network. The small world coefficient is calculated from two other global network metrics:

(1) Network Clustering Coefficient: The clustering coefficient is the average fraction of pairs of an agent's collaborators who have also collaborated with one another. Mathematically [7]:

$$\mathcal{CC} = \frac{3 \times \textit{number of triangles in the network}}{\textit{number of connected triples of vertices}}.$$

Here a "triangle" is a trio of agents (producers, writers and directors), each of whom is connected to both of the others, and a "connected triple" is a single agent connected to two others.

(2) Network Average Shortest Path: Let P(a,b) be the set of paths between a given pair of agents a and b. We define the shortest path $\pi(a,b)$ as the one having the lowest number of hops between a and b that belongs to P(a,b). Let Π be the set of the shortest paths between any pair of agents in the network. The network average shortest path is given by:

$$\bar{\pi} = \frac{\sum_i^{|\Pi|} \pi_i}{|\Pi|}.$$

Before introducing the small world coefficient itself, it is worth noting that this network is a projection from a bipartite structure; so the measurements had to be corrected by dividing them with the equivalent random graph counterparts [10]. The small world coefficient is given by:

$$\mathcal{Q} = \frac{\mathcal{CC}}{\bar{\pi}}.$$

The small world coefficient allow us to verify the connectivity and cohesion among the producers, writers and directors. The more a network exhibits characteristic of a small world, the more connected the agents are to each other and connected to agents who know each other through past collaborations. We can access the correlation between this network metric and success by associating movie's success parameters with the small world coefficient from the whole network at the time of the movie's release.

Also considering the network topology at the time the movie was released, we calculate metrics that are related to the team that produced the movie and its relative position in the network. These metrics allow to evaluate the previous experience, degree of interaction and cohesion among the agents. Let τ_m be the team that produced a given movie m with size equals to the cardinality $|\tau_m|$. Based on Uzzi [14], we define the following metrics:

(1) Average Previous Team Experience: Let τ_m^2 the binary Cartesian product of the team that produced a movie m. Let $T_E(a,b,c)$ be the number of movies produced by the agents a and b, before the current movie c. The average number of movies each pair of team members jointly produced before, considering all possible pairs in the team, i.e, the Average Team Experience, is given by:

$$\bar{T_E}(\tau_c) = \frac{\sum_{\forall (a,b) \in \tau_c^2} T_E(a,b,c)}{|\tau_c^2|}.$$

(2) Average Previous Team Shared Collaborators: Let $T_S(a,b,c)$ be the number of collaborators a pair of team

[7]Historical Currency Converter: http://currencies.apps.grandtrunk.net

[8]CPI Inflation Calculator: http://www.bls.gov/data/inflation_calculator.htm

members have in common, before the current movie c. The Average Number of Shared Collaborators is given by:

$$\bar{T_S}(\tau_c) = \frac{\sum_{\forall (a,b) \in \tau_c^2} T_S(a,b,c)}{|\tau_c^2|}.$$

(3) Average Previous Team Clustering Coefficient: Let $T_{CC}(a,c)$ be the local clustering coefficient[9] of the agent a, before the current movie c. The average previous team clustering is given by:

$$\bar{T_{CC}}(\tau_c) = \frac{\sum_{\forall a \in \tau_c} T_{CC}(a,c)}{|\tau_c|}.$$

(4) Average Previous Team Closeness: The closeness metric indicates how close a given agent is to any other agent in the whole network and it is calculated from the shortest path metric[10]. Let $T_{Cl}(a,c)$ be the closeness metric of the agent a, before the current movie c[11]. The average previous team closeness is given by:

$$\bar{T_{Cl}}(\tau_c) = \frac{\sum_{\forall a \in \tau_c} T_{Cl}(a,c)}{|\tau_c|}.$$

(5) Average Previous Team Betweenness: The betweeness metric indicates the frequency of the shortest paths from any pair of source and destination that pass through the agent a. Let $T_B(a,c)$ be the betweenness metric of the agent a, before the current movie c. The average previous team betwenness is given by:

$$\bar{T_B}(\tau_c) = \frac{\sum_{\forall a \in \tau_c} T_B(a,c)}{|\tau_c|}.$$

Focusing on the individual performance of agents, we also analyze interesting non-topological metrics. These metrics help to evaluate the individual experience and track record from members of the team.

(1) Average Previous Individual Experience: Let $I_E(a,c)$ be the number of movies produced by the agent a, before the current movie c. The average number of of movies previously produced by team members before the current movie, i.e, the Average Individual Experience, is given by:

$$\bar{I_E}(\tau_c) = \frac{\sum_{\forall a \in \tau_c} I_E(a,c)}{|\tau_c|}.$$

(2) Average Previous Team Rating: Let $T_R(a,c)$ be the average rating of the movies produced by the agent a, before the current movie c. The Average Team Rating is given by:

$$\bar{T_R}(\tau_c) = \frac{\sum_{\forall a \in \tau_c} T_R(a,c)}{|\tau_c|}.$$

(3) Average Previous Team Gross Income: Let $T_G(a,c)$ be the average gross income of the movies produced by the agent a, before the current movie c. The Average Team gross income is given by:

$$\bar{T_G}(\tau_c) = \frac{\sum_{\forall a \in \tau_c} T_G(a,c)}{|\tau_c|}.$$

[9]The set of triangles and triples are restrict to the agent neighborhood.

[10]Closeness(a) = $\frac{1}{\sum_{\forall i} \pi(a,i)}$

[11]This metric is calculated considering the whole network, but the team metric is restricted to the agents in the current movie c.

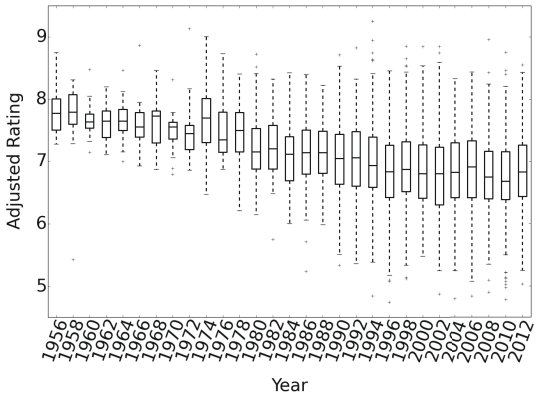


Figure 1. Movie rating distribution per year.

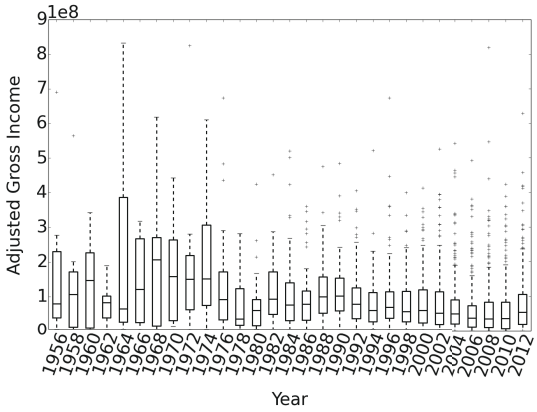


Figure 2. Movie gross income distribution per year.

V. Experimental Evaluation

Before focusing on our main analysis, we present how rating, gross income and small world coefficient evolve over time in our database. After, we discuss how topological and non-topological metrics impact the success parameters considered in our work (*rating* and *gross income*).

A. Historical Evolution of the Network

Figure 1 shows the rating distribution from 1955 to 2013. Average rating for movies decreased almost one point, in average, over the years (from $\approx$ 8 to $\approx$ 7). Interestingly, rating has been spread over the years (for instance, in 2013, the minimum rate is 5.03 and the maximum rate is 8.38). These results suggest that the average movie quality decreased over the years, from the public point of view. This effect could also be due to selection bias: possibly the bad

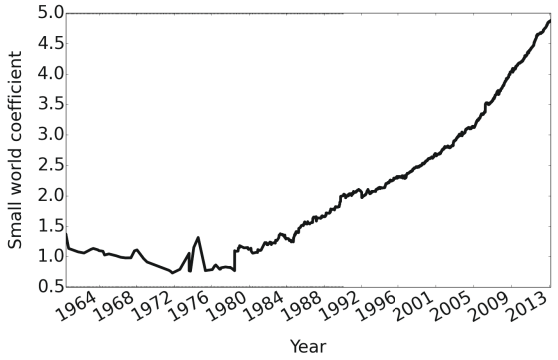


Figure 3. Evolution of the small world coefficient.

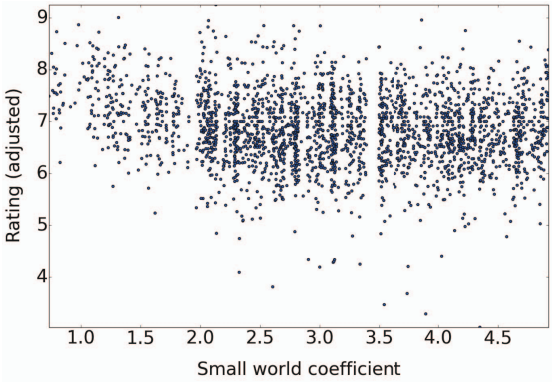


Figure 4. Rating and small world coefficient.

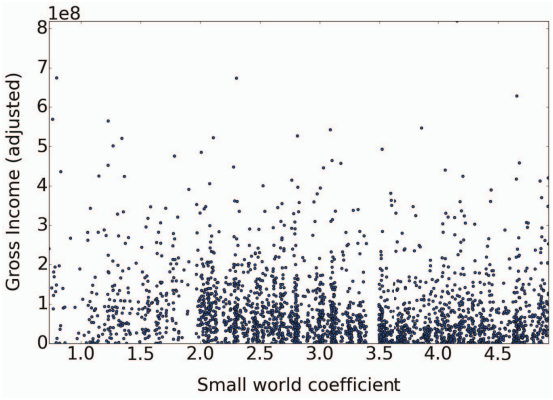


Figure 5. Gross income and small world coefficient.

movies produced a long time ago are being ignored by the public, receiving no or few ratings, whereas newer movies all receive ratings, regardless if they are good or bad.

Figure 2 depicts the gross income achieved by the movies we analyzed. Similarly to the rating, gross income also shows a decreasing pattern. Even considering only movies eligible for the TOP 250 list, there are a set of movies with much higher gross income. These results indicate the high heterogeneity of the financial success in the film industry. The movie with highest gross income in 2013 earned $409 million.

The small world coefficient behavior is presented in Figure 3. From 1961 to 1980, small world coefficient is low. In these years, teams are very spread over the network, with very few links that do exist between them. However, since 1980 the coefficient grows monotonically, indicating high connectivity and cohesion among teams in the network. The network is getting more and more closely knit, with a large number of third-party-in-common relationships. As discussed by Uzzi and Spiro [14], the increase in the level of connections among teams can add the necessary level of credibility needed to facilitate the spread of potentially fresh but unfamiliar creative material by the producers in the network.

B. Topological Metrics

We turn our attention to better understanding how network characteristics impact the success of movies. First, we discuss the small world coefficient. Figures 4 and 5 show the results, for the rating and gross income metrics, respectively. As a global metric, its value does not depend on a specific team but on the whole network. We calculate the coefficient for a movie considering the whole network at the time the movie is released. For the movie rating (Figure 4), it is interesting to highlight that as the small world coefficient increases, the overall rating tend to decrease. There are some movies with rating below 6 (for $\mathcal{Q} > 2$). Our results tend to follow the conclusions made by Uzzi and Spiro [14] that claim that high connectivity may homogenize the pool of creative material, interfering in the production of good movies. However, it is worth nothing that we did not find the strong correlation found by Uzzi and Spiro in their work. For the gross income metric (Figure 5), small world coefficient does not reveal any tendency of correlation.

Figures 6 and 7 present the correlation between the average previous team experience and rating and gross income, respectively. First, most of the movies have low values for average previous team experience. We observe that movies with high values for this metric are less likely to receive a high rating or achieving a large gross income. We can suppose that people who always work together are less likely to have new ideas or courage to innovate. This finding agrees with many works in the literature: new collaborators are highly likely to bring new ideas, resulting in a movie with high potential of achieving success.

The results of the average previous shared collaborators ($\bar{T_S}(\tau_c)$) behavior corroborate the affirmation about the correlation between novelty and success. Figures 8 and 9 depict the results. Teams with the highest values for $\bar{T_S}(\tau_c)$ tend to think in the same way without bringing novelty to the movies that they are producing. Then, these teams tend

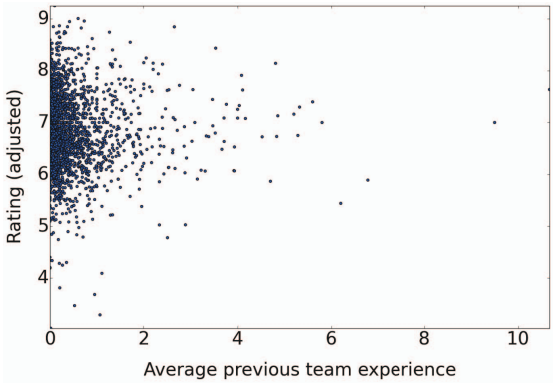


Figure 6. Rating and average previous team experience.

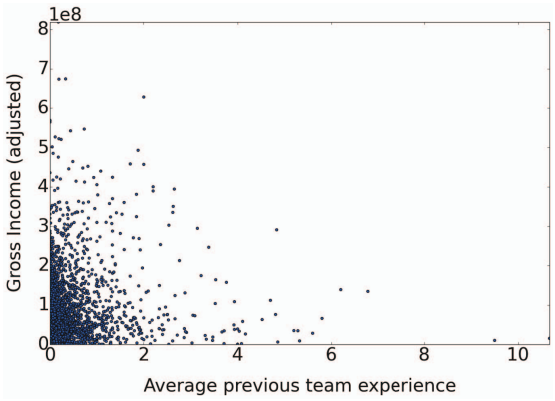


Figure 7. Gross income and average previous team experience.

to be less successful. Teams with the lowest values for the metric are the ones who generate the exceptional ratings and gross income.

Although it also represents the level of team cohesion, the average previous team clustering coefficient seems to be uncorrelated to rating or gross income, as shown in

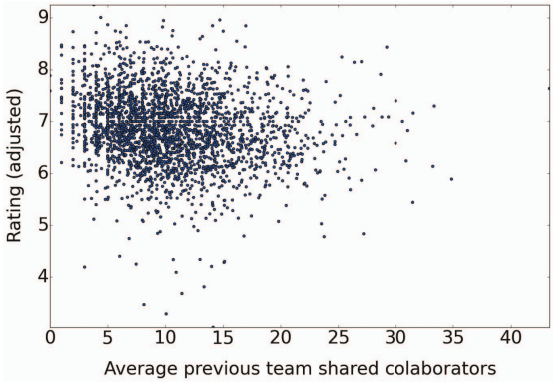


Figure 8. Ratings and average previous shared collaborators.

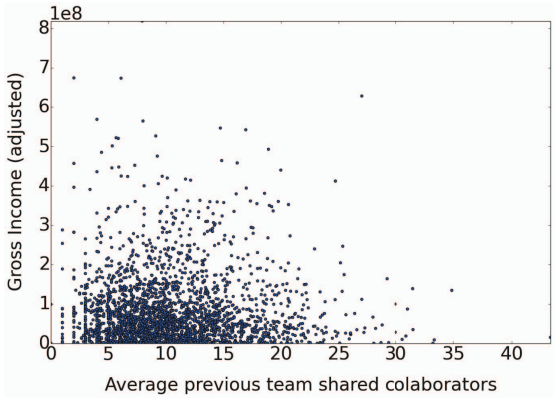


Figure 9. Gross income and average previous shared collaborators.

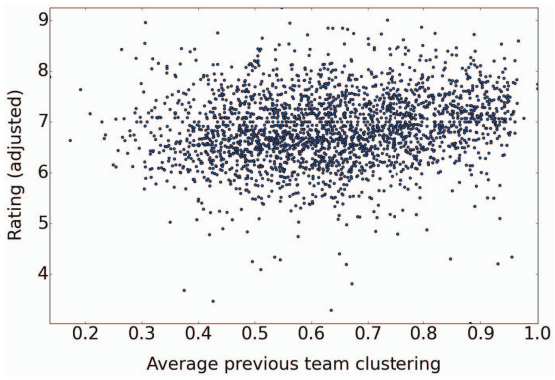


Figure 10. Rating and Average Previous Team Clustering

Figures 10 and 11. As previously discussed, it is important to have some level of previous collaboration to achieve success. However, the number of triangles does not seem to influence the movie success.

Let us focus on the average previous team closeness ($\bar{T_{Cl}}(\tau_c)$), presented in Figures 12 and 13. Movies with

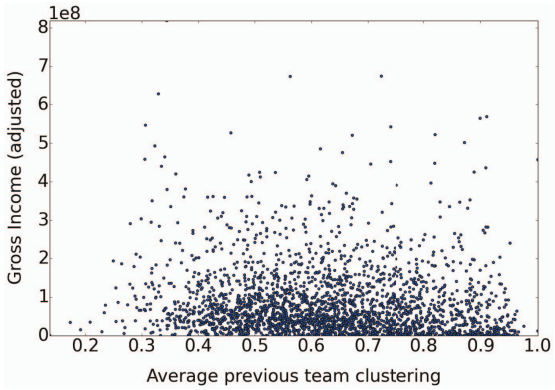


Figure 11. Gross income and Average Previous Team Clustering

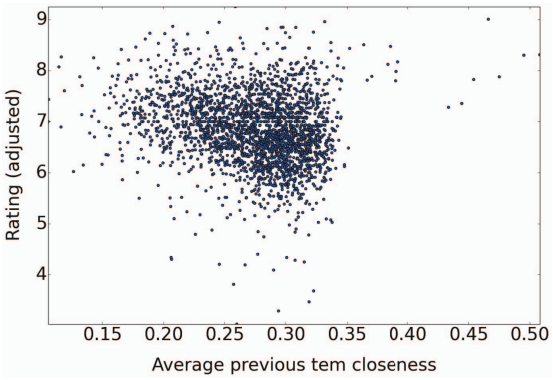


Figure 12. Rating and average previous team closeness.

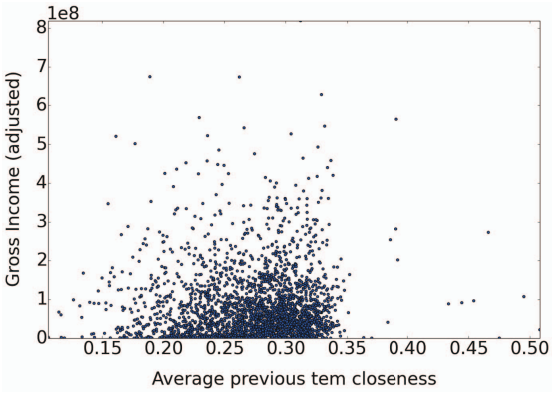


Figure 13. Gross income and averageprevious team closeness.

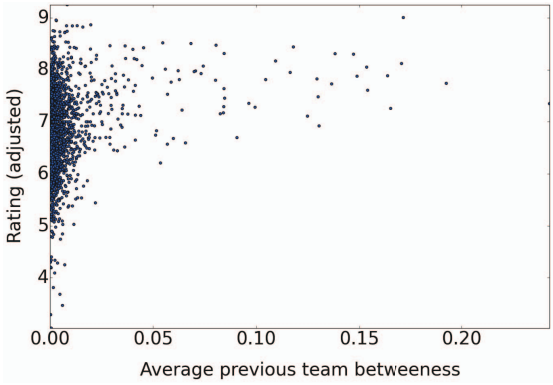


Figure 14. Rating and average previous team betweenness.

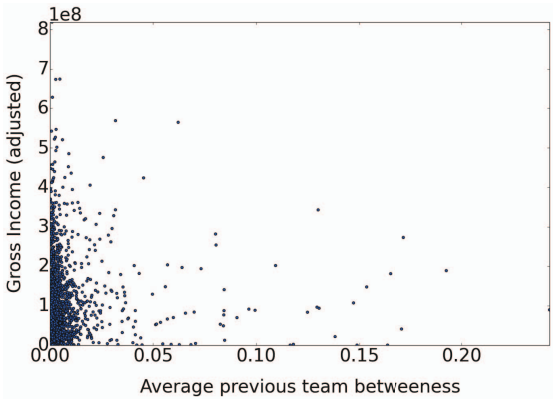


Figure 15. Gross income and average previous team betweeness.

intermediate values of $\bar{T_{Cl}}(\tau_c)$ tend to receive ratings lower than 7.0. Only some outliers with very high values for $\bar{T_{Cl}}(\tau_c)$ receive better ratings. In the other hand, teams with intermediate values of $\bar{T_{Cl}}(\tau_c)$ produce movies with amass higher gross income. We can suppose that, producers who are a few step from successful producers tend to attract the public attention inducing them to watch the movie, increasing the gross income. However, after watching these movies, the public acclamation is not that high, explaining the low rating.

Figures 14 and 15 show the results for the average previous team betweenness ($\bar{T_B}(\tau_c)$). For small values of$\bar{T_B}(\tau_c)$ the betweenness is correlated neither to rating nor gross income. However, for the rating score, values of $\bar{T_B}(\tau_c) > 0.05$ attract the rating values to values around 7.5.

C. Non-Topological Metrics

Next, we focus on analyzing the correlation between non-topological metrics and success. First, let us focus on the average previous individual experience. Movies with the highest rating scores and gross incomes tend to be produced by teams in which directors, writers and producers have little experience in the past. Figures 16 and 17 show the results. Interestingly, teams with much experience in the past tend to have less success. This result counter intuitively shows that teams that already produced many movies before in fact are less likely to produce movies with high public acclamation and high gross income. We may suppose that teams with less experience are mostly composed by young people who are not afraid to innovate. Of course, we can not generalize our conjecture. There are many producers with large experience who frequently produce movies that achieve tremendous success.

Average previous team rating is the metric that best correlates and explains movies' rating and gross income. Figures 18 and Figure 19 present the results. There is a clear correlation between the metric and the current rating. Moreover, most of the movies with amassed the highest gross income were produced by teams with average previous rating above 6.0. Average previous team gross income, instead, seems do not be correlated either to the movie's rating or gross income, as shown in Figures 20 and 21.

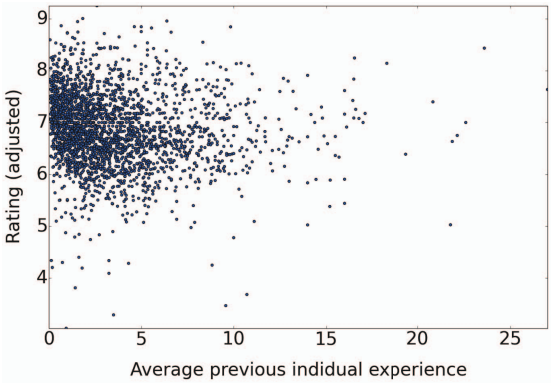


Figure 16. Rating and average individual experience

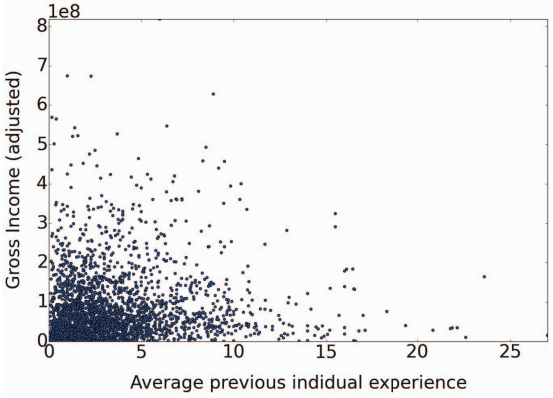


Figure 17. Gross income and average individual experience

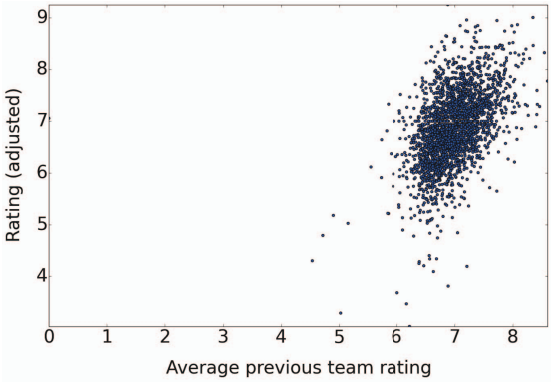


Figure 18. Rating and average previous team rating

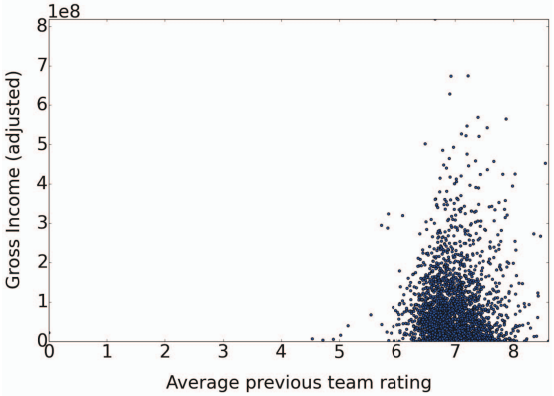


Figure 19. Gross income and average previous team rating.

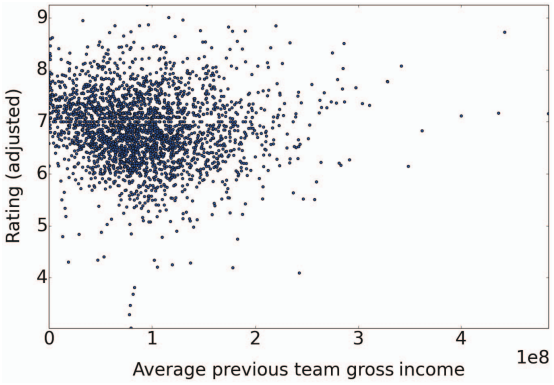


Figure 20. Rating and average previous team gross income.

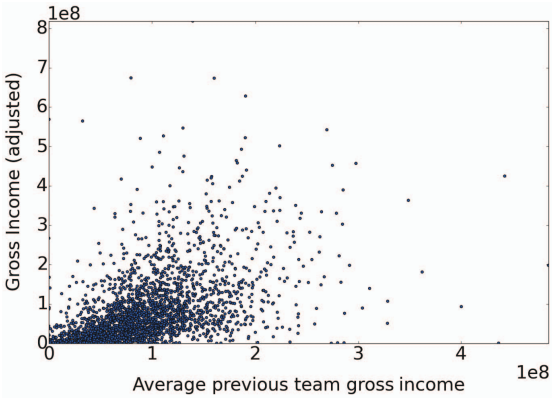


Figure 21. Gross income and average previous team gross income.

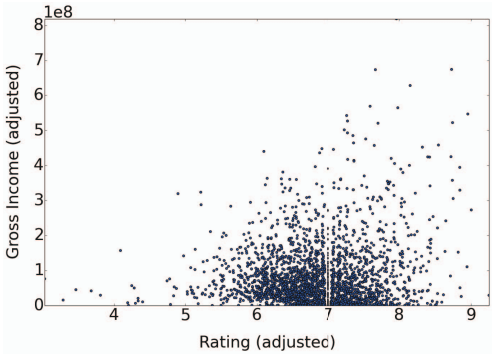


Figure 22. Rating and gross income correlation.

D. Rating versus gross income

An interesting question to explore is whether movie's gross income and ratings are correlated. Figure 22 shows that movies with rating above 6.0 tend to achieve the highest gross incomes in our database. Then, rating and gross income tend to be correlated considering movies in the data we analyzed.

VI. Conclusion

We have presented a broad study on how topological and non-topological metrics of the network of directors, producers and writers impact the success of a movie produced by this team of people. Our findings are very interesting. Non-topological metrics, such as team's average previous rating, centered on the individual or on the team itself had more clearly correlation to the success metrics. Some topological metrics showed to be weakly correlated to the movie success. Interestingly, we found that teams with too much past experience perform worse than teams with fresher agents, reinforcing the assumption that novelty helps to form successful teams.

Besides giving some insights of the correlation between team formation and success, our results are important to show that the team success in film industry is not that simple to characterize, and more elaborate metrics have to be considered. For instance, we believe that the team success can be explained by considering jointly individual and team characteristics or by a more elaborate combination of topological metrics. As an improvement, other strategies can be used to aggregate value from the team members, for example the maximum value or the harmonic mean. Furthermore, other network metrics can be used, such as Burt's structural hole index. We plan to address such points in future work. Furthermore, we are working on employing more metrics to measure movie success, including popularity in online social networks and ratings from other websites, such as Metacritics and Rotten Tomatoes.

Acknowledgments. This work was partially funded by CAPES, CNPq and FAPEMIG.

References

[1] R. S. Burt. Structural Holes and Good Ideas. *The American Journal of Sociology*, 110(2):349–399, 2004.

[2] Z. Chen and J. Guan. The impact of small world on innovation: An empirical study of 16 countries. *Journal of Informetrics*, 4:97–106, 2010.

[3] A. Elberse. The Power of Stars: Do Star Actors Drive the Success of Movies? *Journal of Marketing*, 71:102–120, Oct. 2007.

[4] J. Kleinberg, S. Suri, É. Tardos, and T. Wexler. Strategic network formation with structural holes. In *Proceedings of the 9th ACM Conference on Electronic Commerce*, pages 284–293. ACM, 2008.

[5] E. Y. Li, C. H. Liao, and H. R. Yen. Co-authorship networks and research impact: A social capital perspective. *Research Policy*, 42(9):1515–1530, 2013.

[6] K. Nemoto, P. Gloor, and R. Laubacher. Social capital increases efficiency of collaboration among wikipedia editors. In *Proceedings of the 22nd ACM conference on Hypertext and hypermedia*, pages 231–240. ACM, 2011.

[7] M. Newman. The Structure and Function of Complex Networks. *SIAM Review*, pages 167–256, 2003.

[8] M. E. Newman. Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality. *Physical review E*, 64(1), 2001.

[9] M. E. Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences*, 98(2):404–409, 2001.

[10] M. E. Newman, S. H. Strogatz, , and D. J. Watts. Network effects: Random graphs with arbitrary degree distributions and their applications. *Phys. Rev. E Stat. Nonlin Soft Matter Phys*, 64(2), 2001.

[11] M. A. Schilling and C. C. Phelps. Interfirm collaboration networks: The impact of large-scale network structure on firm innovation. *Management Science*, 53(7):1113–1126, 2007.

[12] P. V. Singh, Y. Tan, and V. Mookerjee. Network effects: The influence of structural capital on open source project success. *MIS Quarterly*, 35(4), 2011.

[13] D. Stokols, K. Hall, B. Taylor, and R. Moser. The science of team science: overview of the field and introduction to the supplement. *American Journal of Preventive Medicine*, 35(2):77–89, 2008.

[14] B. Uzzi and J. Spiro. Collaboration and creativity: The small world problem. *American journal of sociology*, 111(2):447–504, 2005.

Analyzing the Coauthorship Networks of Latin American Computer Science Research Groups

Juan F. Delgado-Garcia, Alberto H. F. Laender and Wagner Meira Jr.
Computer Science Department
Federal University of Minas Gerais
31270-901 - Belo Horizonte - Brazil
{*jfdgarcia, laender, meira*}*@dcc.ufmg.br*

***Abstract*—In this paper, we analyze the coauthorship networks of Latin American Computer Science research groups from 35 academic institutions in Argentina, Brazil, Chile, Colombia, Cuba, Mexico, Peru, Uruguay and Venezuela. Our analysis is based on data over a period of 20 years collected from DBLP, and aims to know the topological structure of each of these networks and provide a view of how they have evolved over time. Our results show that over the 2004-2013 decade there has been a relevant increase in terms of publications and collaborations in Latin America. We also identify the influential authors in the area according to complex network metrics and analyze the research networks originated from the coauthorships. Despite the increase in all per-country metrics, we observed that there is still a lot to improve, since most of the collaborations happen between just Brazil-Chile and Argentina-Brazil, although there is some growth in the diversity of the collaborations.**

***Keywords*-Latin American; Coautorship Networks; Bibliometrics;**

I. Introduction

According to Newman [1], a social network is a collection of individuals or groups of individuals connected by some kind of relationship that exist among them. Such individuals or groups are called *actors* and their relationships *ties*. This kind of network can be represented by a graph in which nodes denote actors and edges represent a specific tie among them.

A coauthorship network, is a special type of social network in which the actors represent authors and the ties indicate that the authors have coauthored at least one publication together. Due to the large amount of bibliographic data made available today on the Web, coauthorship networks have been widely studied over the past years [1], [2], [3], [4], [5], providing an interesting view of the academic communities behind them. Among the pioneering works, Newman [1] analyzes three scientific communities (Computer Science, Physics and Biomedicine) and presents several structural and topological features of them. Similarly, by mapping the publications from important journals in Mathematics and Neurosciences over a eight-year period (1991-1998), Barabási et al. [6] infer the dynamic and the structural mechanisms that govern the evolution and topology of the coauthiorship networks of these two communities based on several metrics.

In the context of the Computer Science (CS) area, Liu et al. [2] present a study of the Digital Library community based on the coauthorship network derived from its three main conferences. In such study, the authors analyze several aspects of this network, including its main connected components and the clustering coefficient. A similar study has also been carried out for the ACM SIGMOD Conference [5]. A comprehensive study of the CS area as whole based on data from CiteSeer[1] has been done by Huang et al. [3], whereas Menezes et al. [7] assess how the process of knowledge production in the area happens in three different geographic regions of the globe: Brazil, North America (Canada and US) and Europe (France, Great Britain and Switzerland).

Regarding specifically the Brazilian research community, Maia et al. [4] present a detailed analysis of the structural features and the evolution of the coautorship network of SBRC[2] thoughout its 30 years of history. In an attempt to contrast the scientific production of the Brazilian and international CS communities, Freire and Figueiredo [8] compare two coauthorship networks generated from DBLP: a global one, created considering all publications found in that digital library, and its subset that considers only researchers affiliated to Brazilian institutions. Finally, using data from the Brazilian National Research Council Lattes Platform[3], Mena-Chalco et al. [9] present a comprehensive study of the Brazilian scientific community by characterizing and exploring its main coauthorship networks. The study aims at gaining an in-depth understanding of the network structures as well as of the dynamics (social behavior) among the researchers in the eight major Brazilian knowledge areas: agricultural sciences, biological sciences, exact and earth sciences, humanities, applied social sciences, health sciences, engineering, and linguistics, letters and arts.

In this paper, we present an analysis of the coautorship networks of Latin American CS research groups. Our analysis is based on data over a period of 20 years (1994-2013) collected from DBLP [10], and addresses 35 institutions from Argentina, Brazil, Chile, Colombia, Cuba, Mexico, Peru, Uruguay and Venezuela. The main aim of

[1]http://citeseerx.ist.psu.edu/index
[2]Brazilian Symposium on Computer Networks and Distributed Systems
[3]http://lattes.cnpq.br

978-1-4799-6953-1/14 $31.00 © 2014 IEEE
DOI 10.1109/LAWeb.2014.13

our analysis is to know the topological structure of each one of these networks and provide a view of how they have evolved over time. Among our main results, we show that there has been a significant increase in the number of publications in the last decade as well as a consolidation of the research groups in some countries. We also identify the influential authors in the area according to three complex centrality network metrics. Finally, we analyze the research networks originated from the coauthorships.

The rest of this paper is organized as follows. Section II describes LACompNet - The Latin American Computer Science Network, a platform we have constructed to support our analysis, Section III discusses our main results, and Section IV presents our conclusions and summarizes future work.

II. LACompNet: The Latin American Computer Science Network

A coauthorship network can be represented as a graph $G_c = (V_c, E_c)$, where V_c represents a set of authors from a community c (e.g., an institution or a specific research group) and E_c represents a set of edges or coauthorship relations between two or more authors in V_c. In other words, an edge between two vertices indicates that the corresponding authors have coauthored at least one publication. In this context, **LACompNet** is a coauthorship network composed of Computer Science researchers from Latin American countries. Here, we consider LACompNet as a non-directed graph, where each vertex's weight corresponds to the number of publications per author during a period, and each edge's weight corresponds to the number of common publications between the corresponding authors.

Data Gathering. The data gathering process consisted of three steps: (i) determining the list of researchers (authors) of each institution of interest, (ii) collecting data from DBLP, and (iii) creating a relational database.

In the first step, we determined the list of researchers by manually extracting their names from the official websites of the 35 Latin American CS graduate programs considered. We then checked whether there was a DBLP entry for each of these researchers and collected their publication data from there. The current version of LACompNet covers researchers from the following institutions:

Argentina - Universidad de Buenos Aires (UBA), Universidad Nacional de la Plata (UNLP), Universidad Nacional del Centro de la Provincia de Buenos Aires (UNICEN) and Universidad Nacional del Sur (UNS); **Brasil -** Universidade Federal do Rio de Janeiro (UFRJ), Universidade Federal de Minas Gerais (UFMG), Universidade Federal do Rio Grande do Sul (UFRGS), Pontifícia Universidade Católica do Rio de Janeiro (PUC-RIO), Universidade Estadual de Campinas (UNICAMP), Universidade Federal de Pernambuco (UFPE), Universidade de São Paulo (USP) and Universidade de São Paulo at São Carlos (USP-SC); **Chile -** Pontificia Universidad Católica de Chile (PUC-Chile), Universidad de Chile (UCHILE), Universidad Santiago de Chile (USACH), Universidad Técnica Federico Santa Maria (UTFSM) and Universidad de Concepción (UDEC); **Colombia -** Universidad ICESI (ICESI), Pontificia Universidad Javeriana at Cali (PUJ-Cali), Universidad de los Andes (ANDES), Universidad del Valle (UNIVALLE) and Universidad Nacional de Colombia (UNAL); **Cuba -** Universidad de La Habana (UH), Universidad de las Ciencias Informáticas (UCI) and Universidad de Oriente (UO); **Mexico -** Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional (CINVES-TAV), Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM), Universidad Autonoma del Estado de Mexico (UAEMEX) and Universidad Nacional Autonoma de Mexico (UNAM); **Peru -** Universidad Católica San Pablo (USCP); **Uruguay -** Universidad de la Republica (UDELAR); **Venezuela -** Universidad Central de Venezuela (UCV), Universidad de Carabobo (UC), Universidad Simón Bolívar (USB) and Universidad de Los Andes (ULA).

In the second step, we collected data from the DBLP entry of each identified researcher. In particular, we collected data from a 20-year period that ranges from 1994 to 2013. Finally, in the third step, we used a Java crawler to extract the data of interest from the DBLP pages. Then, we used the Simple API for XML (SAX[4]) for parsing the resulting XML files, and populated a relational database (MySQL) to ease data querying and analysis.

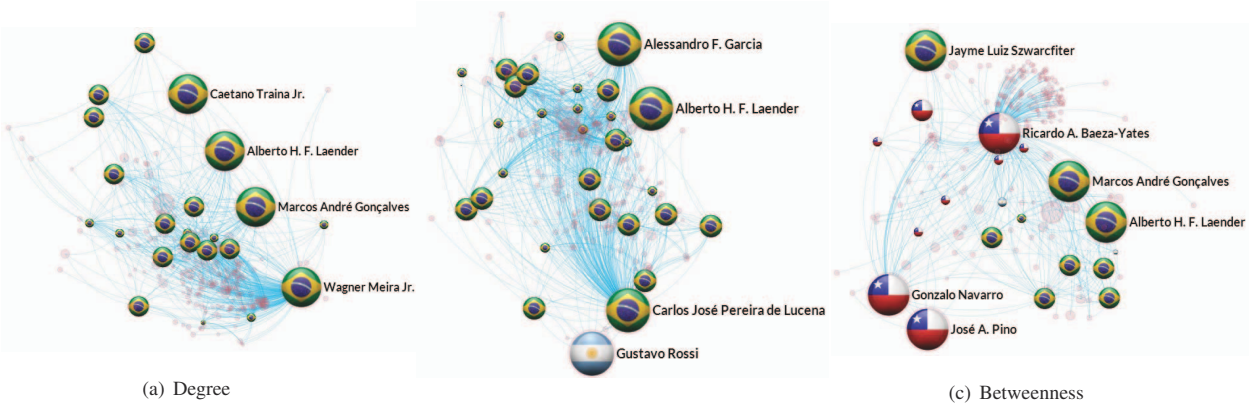
Basic Statistics. Here we present some basic statistics for LACompNet. The current graph is composed of 15601 vertices and 24722 edges. The total number of publications during the period between 1994 and 2013 is 18930 in 2887 venues, including books, journal articles and conference papers. Further, out of the 15601 vertices, 904 correspond to faculty members from the 35 Latin American institutions included in our study and the other 14697 are considered only as coauthors. We notice that 67.80% of them are from Brazilian, Chilean and Argentinian institutions.

Centrality Metrics. Graph centrality metrics are used to analyze the topological structure of a network, as well as to characterize the behavior and evolution pattern across time. In this work we use the *degree*, *closeness* and *betwenness centrality* metrics [11]. Tables I, II and III show, respectively, the list of the top-10 authors for each of these centrality metrics and Figures 1(a), 1(b) and 1(c) show the coauthorship networks of the top researchers according to each metric.

III. Network Analysis

In this section we analyze the coauthorship networks from each country as a whole and for each of the two periods of study (1994-2003 and 2004-2013). The general graph measures of LACompNet are presented in Table IV. Notice that the degree assortative coefficients are negative for all countries, which means that high-degree nodes tend

[4]http://www.saxproject.org/



(a) Degree

(b) Closeness

(c) Betweenness

Figure 1. Coauthorship networks of the top researchers according to each graph centrality metric.

Table I
TOP-10 AUTHORS BASED ON DEGREE CENTRALITY.

Author	Institutions	Degree
Wagner Meira Jr.	UFMG	0.0155
Ricardo A. Baeza-Yates	UCHILE	0.0151
Carlos José Pereira de Lucena	PUC-RIO	0.0133
Marcos André Gonçalves	UFMG	0.0130
Luigi Carro	UFRGS	0.0123
Alessandro F. Garcia	PUC-RIO	0.0115
Jano Moreira de Souza	UFRJ	0.0112
Silvio Romero de Lemos Meira	UFPE	0.0109
Jussara M. Almeida	UFMG	0.0107
Carlos A. Coello Coello	CINVESTAV	0.0106

Table II
TOP-10 AUTHORS BASED ON CLOSENESS CENTRALITY.

Author	Institutions	Closeness
Carlos José Pereira de Lucena	PUC-RIO	0.2453
Alberto H. F. Laender	UFMG	0.2425
Ricardo A. Baeza-Yates	UCHILE	0.2387
Nivio Ziviani	UFMG	0.2386
José Carlos Maldonado	ICMC-USP	0.2370
Simone Diniz Junqueira Barbosa	PUC-RIO	0.2339
Jussara M. Almeida	UFMG	0.2331
Wagner Meira Jr.	UFMG	0.2324
Marcos Andre Gonçalves	UFMG	0.2322
Thaís Vasconcelos Batista	UFRN	0.2312

Table III
TOP-10 AUTHORS BASED ON BETWEENNESS CENTRALITY.

Author	Institutions	Betweenness
Ricardo A. Baeza-Yates	UCHILE	0.0986
Carlos José Pereira de Lucena	PUC-RIO	0.0600
José Carlos Maldonado	ICMC-USP	0.0412
Jayme Luiz Szwarcfiter	UFRJ	0.0352
Wagner Meira Jr.	UFMG	0.0344
Antonio Alfredo Ferreira Loureiro	UFMG	0.0316
Carlos A. Coello Coello	CINVESTAV	0.0308
Jorge Urrutia	UNAM	0.0306
Sergio F. Ochoa	UCHILE	0.0295
Alberto H. F. Laender	UFMG	0.0289

Table IV
GRAPH MEASURES PER COUNTRY.

Country	Inst.	Graph Size $\|E\|$	Deg. Ass. Coeff.	Avg Clust.	Cliques [+]	LCC [*]
Argentina	4	1839	-0.371	0.247	1273	964
Brazil	8	17567	-0.485	0.286	11756	10401
Chile	5	2674	-0.306	0.195	1982	1755
Colombia	5	553	-0.448	0.105	479	252
Cuba	3	291	-0.310	0.217	189	54
Mexico	4	1729	-0.291	0.096	1515	1278
Peru	1	61	-0.597	0.047	57	22
Uruguay	1	379	-0.386	0.158	271	232
Venezuela	4	677	-0.329	0.207	505	289

[+] Cliques: Subsets of the vertex set $C \subseteq V$, such that for every two vertices in C, there exists an edge connecting the two.
[*] LCC: Largest Connected Component

to attach to low-degree nodes. In our case, this indicates that senior researchers frequently publish together with younger researchers and students. Observing the average clustering, we can confirm the trend regarding a larger number of collaborations in Brazil, contrasting to the fewest collaborations in Peru. Such differences w.r.t. critical mass in each country are confirmed by the number of cliques and the size of the largest connected components.

Table V shows an average relative increase of 57.50% w.r.t. the number of authors (including researchers and collaborators) and 69.11% w.r.t. the number of coauthorships. Colombia, Cuba, Peru and Uruguay are the countries that presented the largest increase in both metrics, since they are the countries for which we considered the smallest number of institutions and had the fewest researchers in the first period of analysis (1994-2003). On the other hand, Brazil presented the smallest relative increase, as a consequence of our study having considered only the top eight institutions in

Brazil (levels 6 and 7 according to CAPES[5], the Brazilian Ministry of Education agency in charge of graduate programs), which were already quite consolidated during both periods of study.

Table V
COAUTHORSHIP NETWORK GRAPH STATISTICS.

Country	1994 - 2003		2004 - 2013		Rel. Inc. %	
	Vertices	Edges	Vertices	Edges	Vertices	Edges
Argentina	1076	1450	1292	1830	20.07	26.20
Brazil	10113	16277	10560	17552	4.42	7.83
Chile	1661	2095	1958	2667	17.88	27.30
Colombia	248	259	505	545	103.62	110.42
Cuba	94	100	209	284	122.34	184
Mexico	1322	1455	1537	1724	16.26	18.48
Peru	27	25	65	61	140.74	144
Uruguay	182	219	290	379	59.34	73.05
Venezuela	408	507	542	663	32.84	30.76
Average	1681.22	2487.44	1884.22	2856.11	57.50	69.11

Research Networks. In order to assess the level of collaboration among the researchers in each network, here we present an analysis of the research networks that arose from the coauthorships and discuss how they evolved across time. We define a research network as a group of authors who have published together at least five papers in a decade [12]. We determined such networks by mining maximal sets [11] of authors. Table VI shows both the number of groups and their average size per country and decade. As we can see, all countries presented an increase w.r.t both indicators. The increase in the number of groups demonstrate that there is an increasing research density in the region, while the increase in the average size of the groups shows that researchers are cooperating more and there is a growing critical mass in the area. It is also interesting to notice that such trend appears for almost all countries regardless the number of papers published and number of research groups. Such evaluation also shows the increase of the critical mass in Latin America in terms of active countries, as demonstrated by the appearance of research groups in Colombia, as well as significant increases w.r.t. the number of groups in other countries. The only exception is Peru, which experienced a ten fold increase w.r.t. papers published after 2003, but there was no research group that published more than four papers. In fact, just one research group published four papers, four groups published two papers, and the other groups published just one paper each.

International Collaboration. Here we analyze the cooperations between Latin American countries as materialized by papers that contain authors from more than one country. We start by presenting, in Table VII, the number of papers published by authors from institutions located in a given country in the two periods of analysis. All countries experienced increases in the number of papers published. In Brazil and Argentina the increase was more than 280% in each country, in Mexico 181%, and in Chile 423%. Several countries experienced quite impressive increases: Colombia (1944%), Cuba (514%), Peru (933%), and Uruguay (510%). However, it is remarkable that the number of papers produced by these countries till 2003 was very small, being at most 29 papers in 10 years. Finally, Venezuela showed the worst increase (148%), maybe as a consequence of recent political changes there.

[5]http://www.avaliacaotrienal2013.capes.gov.br/

Table VI
GROUP ANALYSIS BY COUNTRY.

Country	1994 - 2003		2004 - 2013		Total
	Groups	Avg Size	Groups	Avg Size	
Argentina	24	2.17	108	2.72	132
Brasil	162	2.33	1060	2.60	1262
Chile	15	2.47	158	2.49	173
Colombia	0	0.00	23	2.43	23
Cuba	1	3.00	7	2.71	8
Mexico	25	2.48	96	2.54	121
Uruguay	1	2.00	16	2.50	17
Venezuela	6	2.00	24	2.38	30

Table VII
DISTRIBUTION OF PAPERS BY COUNTRY.

Country	1994 - 2003	2004 - 2013	Total
Argentina	307	1207	1514
Brazil	2654	10157	12811
Chile	396	2069	2465
Colombia	16	327	343
Cuba	21	129	150
Mexico	456	1280	1736
Peru	3	31	34
Uruguay	29	177	206
Venezuela	155	384	539

We then proceed and check the number of papers that were written by authors from more than one country, which is shown in Table VIII. A very first observation is that all cooperative work involved just two countries. We could not find in our dataset papers from authors who work on more than two countries. We also present, for each decade, the relative importance of these papers considering the total number of papers published by the countries being considered. Given that the total number of papers published by country A is p_A, country B is p_B, and both is p_{AB}, the relative importance of p_{AB} is given by $p_{AB}/(p_A + p_B - pAB)$.

We can observe that, before 2004, there were just two significant cooperations in Latin America, namely Argentina-Brazil and Brazil-Chile. We found two other cases, but they were not significative, although the cooperation Brazil-Peru was responsible for two of the three publications from Peru in that period. When we look at the last ten years, we can see that several new cooperations arose, but none of them was as significative as those from the previous decade, being most of them in the single digits in terms of number of papers. It is also remarkable that we had a

Table VIII
DISTRIBUTION OF JOINT PAPERS BY COUNTRY GROUPS.

Country	1994 - 2003		2004 - 2013		Total
	Papers	% Total	Papers	% Total	
Argentina, Brazil	42	1.49	47	0.44	89
Argentina, Chile	0	0.00	10	0.32	10
Argentina, Colombia	0	0.00	1	0.07	1
Argentina, Mexico	0	0.00	1	0.04	1
Brazil, Chile	19	0.65	26	0.23	45
Brazil, Colombia	0	0.00	7	0.07	7
Brazil, Peru	2	0.08	1	0.01	3
Brazil, Uruguay	0	0.00	4	0.04	4
Brazil, Venezuela	2	0.07	3	0.03	5
Chile, Colombia	0	0.00	1	0.04	1
Chile, Mexico	0	0.00	3	0.09	3
Colombia, Venezuela	0	0.00	1	0.15	1
Cuba, Mexico	0	0.00	5	0.36	5

reduction in the relative importance of the two cooperations from the previous decade, although the absolute number of papers increased (36% between Brazil and Chile and 11% between Argentina and Brazil). It is also worth noting new cooperative work between Argentina and Chile.

IV. CONCLUSIONS

In this paper we studied the coauthorship networks of CS research groups from 35 Latin American institutions. The study is based on data from DBLP and spans the period from 1994 to 2013. Our analysis shows that there has been a significant increase in the number of publications in the last decade, in particular when we consider countries such as Colombia, Uruguay and Venezuela, with less research tradition in the area. We also observed a consolidation of research groups in other countries such as Argentina, Brazil, Chile and Mexico.

We also identified the influential authors in the area, according to the centrality metrics considered, which show a predominance of Brazilian, Chilean and Mexican researchers that are in traditional research centers and were able to establish research groups. These findings may be useful for strenghtening the existing networks and fostering new collaborations with the researchers located in priviledged network locations. We also analyzed research networks that emerged from the coauthorships (i.e., groups of authors who published consistently together) and found that the number and size of such groups increased in almost all countries, showing a clear densification process. Regarding international collaboration, we found that there is still a lot to improve, since most of the collaborations happen between just Brazil-Chile and Argentina-Brazil, although there is some growth in the diversity of the collaborations.

As future work, we intend to study in greater detail how the networks are formed and to investigate the impact of the research topics on the evolution of CS research in Latin America.

ACKNOWLEDGMENTS

This work is partially funded by InWeb (MCT/CNPq/FAPEMIG grant 573871/2008-6), and by the authors' individual grants from CAPES, CNPq and FAPEMIG.

REFERENCES

[1] M. E. Newman, "The structure of scientific collaboration networks," *PNAS*, vol. 98, no. 2, p. 404, 2001.

[2] X. Liu, J. Bollen, M. L. Nelson, and H. Van de Sompel, "Co-authorship networks in the digital library research community," *Information Processing & Management*, vol. 41, no. 6, pp. 1462–1480, 2005.

[3] J. Huang, Z. Zhuang, J. Li, and C. L. Giles, "Collaboration over Time: Characterizing and Modeling Network Evolution," in *Proc, of WSDM*, Stanford, CA, USA, 2008, pp. 107–116.

[4] G. Maia, P. O. S. V. de Melo, D. L. Guidoni, F. S. H. Souza, T. H. Silva, J. M. Almeida, and A. A. F. Loureiro, "On the analysis of the collaboration network of the Brazilian symposium on computer networks and distributed systems - 30 Editions of history," *J. Braz. Comp. Soc.*, vol. 19, no. 3, pp. 361–382, 2013.

[5] M. A. Nascimento, J. Sander, and J. Pound, "Analysis of SIGMOD's Co-authorship Graph," *SIGMOD Record*, vol. 32, no. 3, pp. 8–10, 2003.

[6] A. L. Barábasi, Z. Néda, E. Ravasz, A. Schubert, and V. T, "Evolution of the social network of scientific collaborations," *Physica A: Statistical Mechanics and its Applications*, vol. 311, no. 3-4, pp. 590–614, 2002.

[7] G. V. Menezes, N. Ziviani, A. H. F. Laender, and V. A. F. Almeida, "A Geographical Analysis of Knowledge Production in Computer Science," in *Proc. of WWW*, Madrid, Spain, 2009, pp. 1041–1050.

[8] V. P. Freire and D. R. Figueiredo, "Ranking in collaboration networks using a group based metric," *J. Braz. Comp. Soc.*, vol. 17, no. 4, pp. 255–266, 2011.

[9] J. P. Mena-Chalco, L. A. Digiampietri, F. M. Lopes, and R. M. C. Junior, "Brazilian Bibliometric Coauthorship Networks," *JASIST*, vol. 66, no. 7, 2014.

[10] M. Ley, "DBLP: Some Lessons Learned," *Proc. of VLDB*, vol. 2, no. 2, pp. 1493–1500, 2009.

[11] M. J. Zaki and W. Meira Jr, *Data Mining and Analysis: Fundamental Concepts and Algorithms*. Cambridge University Press, 2014.

[12] J. F. Delgado-Garcia, A. H. F. Laender, and W. Meira Jr., "A Preliminary Analysis of the Scientific Production of Latin American Computer Science Research Groups," in *Proc. of AMW*, Cartagena de Indias, Colombia, 2014.

Publishing and querying government multidimensional data using QB4OLAP

M. Bouza, B. Elliot, and L. Etcheverry
Instituto de Computación, Facultad de Ingeniería
UdelaR Montevideo, Uruguay
lorenae@fing.edu.uy

Alejandro A. Vaisman
Instituto Tecnológico de Buenos Aires
Buenos Aires, Argentina
avaisman@itba.edu.ar

***Abstract*—The web is changing the way in which data warehouses are designed, used, and queried. With the advent of initiatives such as Open Data and Open Government, organizations want to share their multidimensional data cubes and make them available to be queried online. The RDF data cube vocabulary (QB), the W3C standard to publish statistical data in RDF, presents several limitations to fully support the multidimensional model. The QB4OLAP vocabulary extends QB to overcome these limitations, and provides the distinctive feature of being able to implement several OLAP operations, such as rollup, slice, and dice using standard SPARQL queries. In this paper we present QB4OLAP Engine, a tool that transforms multidimensional data stored in relational DWs into RDF using QB4OLAP, and apply the solution to a real-world case, based on the national survey of housing, health services, and income, carried out by the government of Uruguay.**

***Keywords*-Semantic Web, Data warehouses.**

I. Introduction

OLAP (or, more generally, Business Intelligence) (BI) software, produces reports and interactive interfaces that summarize multidimensional data via basic aggregation functions (e.g., counts and averages) over various hierarchical breakdowns of the data into groups, defined in the dimension hierarchies. A lot of academic research and industrial development was carried out throughout the 1990's related to conceptual modelling, query processing and optimization, aggregate precomputation, etc. Since the mid 90's, data warehouses (DW) and BI applications have been built to consolidate enterprise business data, allowing taking timely and informed decisions based on up-to-date consolidated data.

However, the web is changing the way in which data warehouses are designed, used, and queried [1]. With the advent of initiatives such as Open Data[1] and Open Government, organizations want to publish multidimensional data using standards and non-propietary formats.[2] Traditionally, OLAP and BI solutions are commercial tools with proprietary formats. Although in the last decade several Open Source BI platforms have emerged, they still do not provide an open format to publish and share cubes among organizations [2]. In addition, the Linked Data paradigm allows sharing and reusing data in the web by means of semantic web standards [3]. Domain ontologies expressed in RDF (the basic data representation layer) or in languages built on top of RDF like RDF-S or OWL, define a common terminology for the concepts involved in a particular domain.

The RDF data cube vocabulary (QB) is the W3C standard to publish statistical data in RDF, presents several limitations to fully support the multidimensional model [4]. The QB4OLAP vocabulary [4], [5] extends QB by means of a set of RDF terms, organized in an RDF-S ontology. A distinctive feature of QB4OLAP is that several OLAP operations, such as rollup, slice and dice, can be implemented as standard SPARQL queries. This implies that, from multidimensional data published at high granularity levels, users can obtain coarse-grained multidimensional data aggregations without the need of specific infrastructure such as cube servers. This approach is aligned with open government principles, in particular with the *primary* principle which states that data should be published as collected from the source, with the highest possible level of granularity and not in aggregate or modified forms.[3]. In light of the above, in this paper we introduce *QB4OLAP Engine*, a tool that transforms multidimensional data stored in relational DWs into RDF using QB4OLAP, and show how real-world multidimensional data can be published and analyzed over the web, taking the national home and income survey in Uruguay.

After introducing background concepts and a description of QB4OLAP (Section II), we present the QB4OLAP Engine (Section III). In Section IV we describe our application scenario: multidimensional data obtained from surveys in Uruguay, and we show how the resulting RDF model can be queried using SPARQL, the standard query language for RDF. Related work is discussed in Section V. Finally we comment on open challenges and future work (Section VI).

II. Preliminaries

RDF The Resource Description Framework (RDF) [6] is a data model for expressing assertions over resources identified by an universal resource identifier (URI). Assertions are expressed as triples *subject - predicate - object*, where *subject* are always resources, and *predicate* and *object* could

[1] https://okfn.org/opendata/
[2] http://opengovdata.org/
[3] http://opengovdata.org/

978-1-4799-6953-1/14 $31.00 © 2014 IEEE
DOI 10.1109/LAWeb.2014.11

be resources or strings. *Blank nodes* are used to represent anonymous resources or resources without an URI, typically with a structural function, e.g., to group a set of statements. Data values in RDF are called *literals* and can only be *objects*. A set of RDF triples or *RDF dataset* can be seen as a directed graph where *subject* and *object* are nodes, and *predicates* are arcs. Usually, triples representing schema and instance data coexist in RDF datasets. A set of reserved words defined in RDF Schema (called the rdfs-vocabulary) [7] is used to define classes, properties, and to represent hierarchical relationships between them. For example, the triple *(s, `rdf:type`, c)* explicitly states that *s* is an instance of *c* but it also implicitly states that object *c* is an instance of `rdf:Class` since there exists at least one resource that is an instance of *c*. Many formats for RDF serialization exist. The examples presented in this paper use Turtle [8].

SPARQL 1.1 [9] is the current W3C standard query language for RDF. The query evaluation mechanism of SPARQL is based on subgraph matching: RDF triples are interpreted as nodes and edges of directed graphs, and the query graph is matched to the data graph, instantiating the variables in the query graph definition. The selection criteria is expressed as a graph pattern in the `WHERE` clause, composed by *basic graph patterns (BGP)*. The '.' operator represents the conjunction of graph patterns. SPARQL supports aggregate functions and the `GROUP BY` clause, which are relevant to OLAP.

R2RML [10] is a language for expressing mappings from relational databases to RDF datasets, using a customized structure and vocabulary. Both, R2RML mapping documents (written in Turtle syntax) and mapping results, are RDF graphs. The main object of an R2RML mapping is the *triples map*, which is a collection of triples composed of a *logical table*, a *subject map*, and one or more *predicate object maps*. A logical table is either a base table or a view (using the predicate `rr:tableName`), or an SQL query (using the predicate `rr:sqlQuery`). A predicate object map is composed of a predicate map and an object map. Subject maps, predicate maps, and object maps are either constants (`rr:constant`), column-based maps (`rr:column`), or template-based maps (`rr:template`). Templates use brace-enclosed column names as placeholders. Foreign keys are handled through referencing object maps, which use the subjects of another triples map as the objects generated by a predicate-object map . We show examples of R2RML mappings in Section IV-A. R2RML mappings.

Linked Data [3], [11] is a data publication paradigm, based on semantic web standards that aims at publishing and relating data on the web. W3C projects such as the Linking Open Data community (LOD)[4] encourage the publication of open data using the linked data principles, which recommend using RDF as data publication format. LOD site, which consisted of than 500 million RDF links.

Multimensional Data Model In OLAP, data are organized as hypercubes whose axes are *dimensions*. Each point in this multidimensional space is mapped through *facts* into one or more spaces of *measures*. Dimensions are structured in *hierarchies* of *levels* that allow analysis at different levels of aggregation. The values in a dimension level are called *members*, which can also have properties or *attributes*. Members in a dimension level must have a corresponding member in the upper level in the hierarchy, and this correspondence is defined through so-called rollup functions.

QB4OLAP The RDF data cube vocabulary (QB),[5] is used to publish statistical data in RDF. The QB4OLAP vocabulary[6] extends QB to enhance the support of the multidimensional model [4]. Unlike QB, QB4OLAP allows implementing the main OLAP operations, such as rollup, slice, and dice, using standard SPARQL queries. Two different kinds of sets of RDF triples are needed to represent a data cube in QB4OLAP: (i) the *cube schema*, and (ii) the *cube instances*.The former defines the structure of the cube, in terms of dimension levels and measures, but also defines the hierarchies within dimensions and the parent-child relationships among levels. This information is needed to automatically obtain SPARQL queries that implement OLAP operations, for example rollup operator [4]. On the other hand, cube instances are sets of triples that represent level members, facts and measured values. Several cube instances may share the same cube schema.

Figure 1 depicts the QB4OLAP vocabulary, which allows data cubes already published using QB to be represented using QB4OLAP without affecting existing applications. Original QB terms are prefixed with '`qb:`'. Capitalized terms represent RDF classes and noncapitalized terms represent RDF properties. Classes in external vocabularies are depicted in light gray font. QB4OLAP classes and properties (with prefix `qb4o`) are depicted in light-gray background. A data structure definition (DSD) specifies the schema of a data set of the class `qb:DataSet`. The DSD can be shared among different data sets, and has properties (`qb:componentProperty`) for representing dimensions, measures, and attributes, called `qb:dimension`, `qb:measure`, and `qb:attribute`, respectively. Observations (facts) represent points in a multidimensional space. An observation is linked to a value in each dimension of the DSD using instances of `qb:DimensionProperty`. The `qb:concept` property links components to the concept they represent, modeled using the `skos:Concept` class defined in the SKOS vocabulary,[7] also used to define hierarchies by means of `skos:broader` and `skos:narrower`.

[4]http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData

[5]http://www.w3.org/TR/vocab-data-cube/

[6]http://purl.org/qb4olap/cubes

[7]http://www.w3.org/2009/08/skos-reference/skos.html

For instance, the triple `country skos:narrower region` represents a hierarchical relationship where region is at a lower level than country. The `skos:hasTopConcept` property provides an entry point to these concept hierarchies.

As mentioned earlier, a key aspect of QB4OLAP is its ability to structure dimensions as hierarchies of levels. The class `qb4o:LevelProperty` models dimension levels. Level members are instances of the class `qb4o:LevelMember`, and relations between them are expressed using the property `skos:broader`. Dimension hierarchies are defined using the class `qb4o:HierarchyProperty`. The relationship between dimensions and hierarchies is represented via the property `qb4o:hasHierarchy` and its inverse `qb4o:inDimension`. A level may belong to different hierarchies, and in each hierarchy it may have a different parent level. Also, the relationships between level members may have different cardinalities (e.g. one-to-many, many-to-many, etc.). The `qb4o:LevelInHierarchy` class represents pairs of levels and hierarchies using the `qb4o:levelComponent` and the `qb4o:hierarchyComponent` properties respectively. The class `qb4o:HierarchyStep` represents the parent-child relationship between two levels in a hierarchy using the `qb4o:childLevel` and the `qb4o:parentLevel` properties respectively. The `qb4o:cardinality` property allows to represent the cardinality of this relationship using members of the `qb4o:Cardinality` class, and can also be used to represent the cardinality of the relationship between a fact and a level.

QB4OLAP also allows to define level attributes via the `qb4o:hasAttribute` property and aggregate functions via the `qb4o:AggregateFunction` class. The association between measures and aggregate functions is represented using the property `qb4o:aggregateFunction`. This property, together with the concept of component sets, allows a given measure to be associated with different aggregate functions in different cubes. In Section IV we show extracts of a QB4OLAP representation of a data cube.

III. QB4OLAP Engine: a Tool to Transform Relational Data Cubes into RDF

We now describe *QB4OLAP Engine*, our approach to obtain a QB4OLAP representation of a relational representation (ROLAP) of a data cube. After providing an overview of the architecture of the developed solution, we present the key aspects of the main components followed by some implementation details (Section III-D).

A. Design and architecture

QB4OLAP Engine is a tool that takes as input the specification of a multidimensional data cube, and its relational implementation (a set of relational tables) and produces two RDF graphs that use the QB4OLAP vocabulary. One of these graphs represents the *schema of the cube* and the other one an *instance of the cube*. These graphs are stored in a so-called RDF triplestore, which also implements an SPARQL endpoint, in short, a web interface to write and execute SPARQL queries. This allows publishing the cubes on the web, and offering the capability of performing queries over them.

Since there is no standard and machine-processable format to specify multidimensional data cubes, we have chosen the format provided by Pentaho Mondrian open-source OLAP server.[8] The Mondrian schema[9] is an XML document that defines a multidimensional database. It contains a *logical model*, consisting of cubes, dimensions, hierarchies, and members, and a *mapping* of this model onto a physical model. The physical model is the source of the data which is presented through the logical model, which is typically a star schema, implemented as a set of tables in a relational database. In a *star schema*, a fact table is linked through foreign keys to one or more denormalized dimension tables. In a *snowflake schema*, dimension tables are normalized, and a dimension is represented as a collection of tables linked to each other through foreign keys. Snowflake schemas and mixed approaches (like starflake) are also supported by Mondrian schema.

QB4OLAP engine is composed of several modules that tackle each one of the extraction and transformation processes. Figure **??** depicts the data and control flow between these modules, and the interaction with external components.

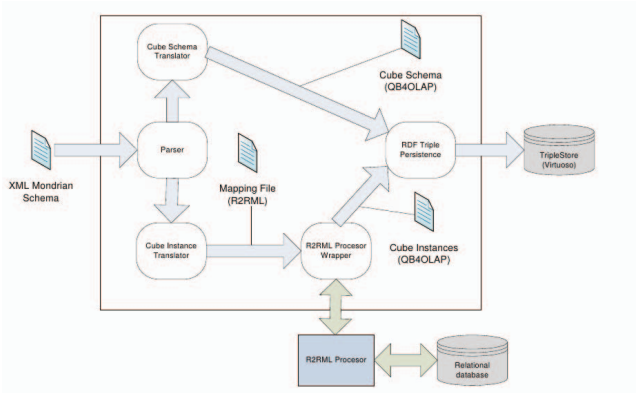


Figure 2: QB4OLAP Engine architecture.

The *parser* component first validates the input XML file that contains the Mondrian schema against its DTD. Then, it extracts the logical model of the cube and the mappings onto the physical model, creating an in-memory representation of this information that will be used throughout all the transformation processes. The *cube schema translator* is responsible for translating the logical model of the data cube

[8] http://community.pentaho.com/projects/mondrian/
[9] http://mondrian.pentaho.com/documentation/schema.php

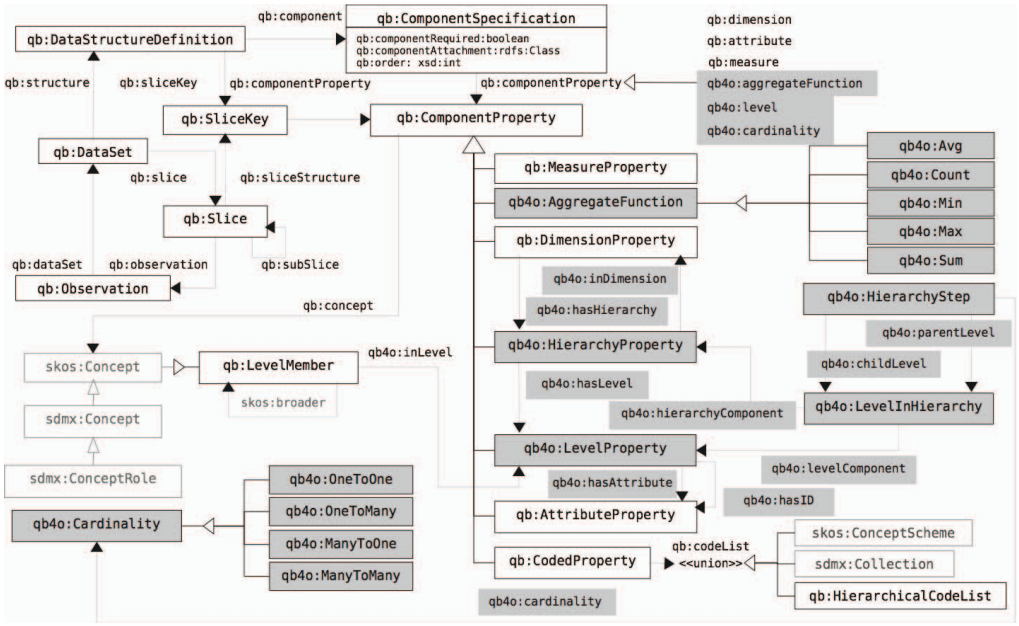


Figure 1: QB4OLAP vocabulary

into RDF, producing as output a set of RDF triples that represent the schema of the cube using QB4OLAP.

The *cube instance translator* is responsible for translating the data in the underlying relational database into RDF triples that represent instances of the data cube. This means generating triples, using QB4OLAP and the terms defined as the schema of the cube, that represent: level members with their corresponding attribute values, parent-child relationships among level members, and fact instances. The task at hand corresponds to the more general problem of providing an RDF view over relational data. Instead of directly generating triples that represent the instances, this component produces a set of R2RML mappings that encode how to produce these instances from the data stored in the physical model of the cube. Then, this set of R2RML mappings can either be used to generate a static set of triples that represent the underlying relational data (*data materialization*) or to provide a non-materialized RDF view of the relational data (*on-demand mapping*). Each of these strategies has a set of well-known advantages and disadvantages. In our current implementation we have chosen to materialize the triples that represent the instance. This decision is mainly based on the static nature of the underlying data (no need for updates). After the mappings are obtained, the *R2RML processor wrapper* interacts with an R2RML processor, which actually builds the RDF triples. Finally, the *RDF triples persistence* module stores all the triples into a triplestore. In the following sections we details the translation process.

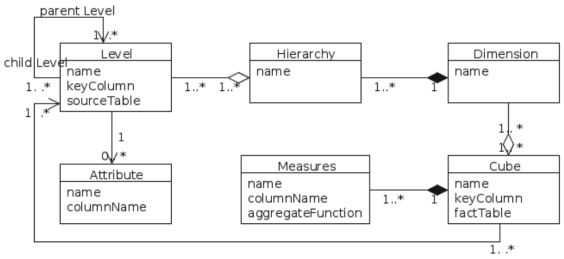


Figure 3: Internal representation of Mondrian schema.

B. Cube Schema Translator

As stated before, the cube schema translator takes as input an in-memory representation of the information contained in the Mondrian schema XML file. Figure 3 shows a diagram of the classes involved in this representation. A *cube* is composed of a set of *dimensions*, each dimension contains one or more *hierarchies*, composed of *levels* which have a set of *attributes*. A cube is also related to a set of levels, one in each of the dimensions involved in the cube. These levels define the granularity of the facts in the cube, and therefore the granularity of the values of the *measures* in the cube. Levels are related to other levels through parent-child relationships. The following restriction applies, and is enforced by XML schema definition. For every pair of levels (l_1, l_2) related via a parent-child relationship, assuming that H_1 and H_2 are the sets of hierarchies to which l_1 and l_2 belong respectively, then $H_1 \cap H_2 \neq \emptyset$ and $|H_1 \cap H_2| = 1$.

Table I: Auxiliary functions

Function signature	Description
getURI(*type*,*name*)	Given the type (ex: level, dimension, etc.) and name of a multidimensional concept, generates or retrieves an absolute URI to identify it.
getPairURI(*URI1*, *URI2*)	Given a pair of URIs, generates or retrieves an absolute URI that identifies the pair.
getTripleMapURI(*name*)	Given the name of a multidimensional concept, generates or retrieves an absolute URI to identify the R2RML TripleMap that will populate the concept.
getTemplate(*name*, *keyColumn*)	Returns a string to be used as a template to build an URI for each level member or fact instance
templateTermMap(*mapType*, *template*, *termType*)	According to *mapType* returns an instance i of rr:SubjectMap, rr:PredicateMap or rr:ObjectMap, where i.rr:template = *template* and i.rr:termType = *termType*
columnTermMap(*mapType*, *column*, *termType*)	According to *mapType* returns an instance i of rr:SubjectMap, rr:PredicateMap or rr:ObjectMap, where i.rr:column=*column* and i.rr:termType = *termType*
constantTermMap(*mapType*, *constant*)	According to *mapType* returns an instance i of rr:SubjectMap, rr:PredicateMap or rr:ObjectMap, where i.rr:constant = *constant*
predicateObjectMap(*predicateMap*, *objectMap*)	Returns an instance i of rr:predicateObjectMap, where i.rr:predicateMap = *predicateMap* and i.rr:objectMap = *objectMap*

Table I presents a list of auxiliary functions used to improve the presentation of the algorithms included in this work. Also, the "." operator is used to retrieve the values of the properties of an object (i.e., if ex: l is a Level, then $l.keyColumn$ retrieves the value of property $keyColumn$). Algorithm 1 describes the schema translation process, which can be decomposed in three steps: (i) processing the dimensions and its structure (lines 1-43), (ii) processing the measures (lines 44-50), and (iii) processing the cube definition (lines 51-57).

C. Cube Instance Translator

The instance translator module also takes as input the internal representation that adheres to the metamodel depicted in Figure 3, but instead of producing a set of RDF triples that represent the instances of the cube, it builds a set of R2RML mappings that will generate those instances. It is worth noting that R2RML mappings are also RDF triples. Algorithm 2 describes the mapping generation process, which can be decomposed in two steps: (i) building R2RML mappings to populate dimensions, which means to create level members and parent-child relationships within them (lines 1-32), and (ii) building R2RML mappings to build facts and measure values (lines 33-57). Auxiliary functions are defined in Table I.

In the first part of the process, for each level in the schema we build mappings that create an RDF representation of the level members stored in the database. Each level member is transformed into an RDF resource. To guarantee the uniqueness of level member URIs, a template is used that takes into account the column stated as the key of each level. Also, each level member is related to the values of each of its attributes using the RDF properties defined in Algorithm 1 (lines 17-22). Finally, we represent parent-child relationships between level members using the `skos:broader` property. In the second part of the process we build a mapping that creates an RDF representation of fact instances. Each fact instance relates a set of level members, one for each dimension in the cube, with a set of measure values.

D. Implementation Details

QB4OLAP Engine has been implemented in Java, using Jena libraries to deal with RDF in-memory representation. MySQL (version 5.5) is used to store the relational representation of the cube, although other RDBMS are easily supported. R2RML Parser[10] is used to process the generated R2RML mappings. Finally, generated RDF triples are stored in Virtuoso Open-Source Edition (version 7.1.0).[11] QB4OLAP Engine source code is available under CC 3.0 BY license.[12]

IV. Case Study: Data from Surveys in Uruguay

In Uruguay, the *Encuesta Continua de Hogares (ECH)* is a continuous survey commissioned by the *Instituto Nacional de Estadística (INE)*, the national office for statistics. It collects information about people's housing, access to health services, employment and income indicators, among others. Since 1968 the EHC is carried out in the capital city (Montevideo); in 1981 it started to cover all the cities in the country, and since 2006 it covers all the territory, including rural areas. This survey is applied over a sample of the target population and INE publishes the sets of records containing information on individual respondents and metadata about ECH, which can be found in Uruguay's government open data catalog.[13]

Among the many data cubes are available, we will focus on a cube that allows to analyze two measures: the number of people (#people) and the number of homes (#homes), according to three dimensions that represent: (a) the comfort level of the home (ComfortLevel), (b) its geographical location (Geography), and (c) the time of the survey (Time). The comfort level (LOW, MEDIUM-LOW, MEDIUM-HIGH, and HIGH) summarizes the presence or absence of different services and appliances such as: internet connection, computer, car, among others. The Geography dimension is organized into levels Neighborhood, City, State, and Region; the Time dimension in Month, Quarter and Year.

[10]https://github.com/nkons/r2rml-parser
[11]http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/
[12]https://code.google.com/p/publishing-multidimensional-data/
[13]https://catalogodatos.gub.uy/dataset?tags=INE

Algorithm 1 Creating the data structure definition of a cube in QB4OLAP from a cube in Mondrian Schema

Input: C_M: internal representation of cube C obtained from Mondrian schema.
Output: C_{RDF}: data structure definition of cube C in QB4OLAP.

$C_{RDF} \leftarrow \emptyset$
$c_U = getURI(cube, C.name)$
$C_{RDF} \leftarrow C_{RDF} \cup \{c_U$ a qb:DataStructureDefinition$\}$
let D be the set of dimensions in C_M
for all $d \in D$ **do**
  $d_U = getURI(dimension, d.name)$
  $C_{RDF} \leftarrow C_{RDF} \cup \{d_U$ a qb:DimensionProperty$\}$
  let H be the set of hierarchies in dimension d
  for all $h \in H$ **do**
    $h_U = getURI(hierarchy, h.name)$
    $C_{RDF} \leftarrow C_{RDF} \cup \{h_U$ a qb4o:HierarchyProperty$\}$
    $C_{RDF} \leftarrow C_{RDF} \cup \{h_U$ qb4o:inDimension $d_U\}$
    $C_{RDF} \leftarrow C_{RDF} \cup \{d_U$ qb4o:hasHierarchy $h_U\}$
    let L be the set of levels in hierarchy h
    for all $l \in L$ **do**
      $l_U = getURI(level, l.name)$
      $C_{RDF} \leftarrow C_{RDF} \cup \{l_U$ a qb4o:LevelProperty$\}$
      $C_{RDF} \leftarrow C_{RDF} \cup \{l_U$ qb4o:inDimension $d_U\}$
      $C_{RDF} \leftarrow C_{RDF} \cup \{l_U$ qb4o:inHierarchy $h_U\}$
      $C_{RDF} \leftarrow C_{RDF} \cup \{h_U$ qb4o:hasLevel $l_U\}$
      let A be the set of attributes in level l
      for all $a \in A$ **do**
        $a_U = getURI(attribute, a.name)$
        $C_{RDF} \leftarrow C_{RDF} \cup \{a_U$ a qb:AttributeProperty$\}$
        $C_{RDF} \leftarrow C_{RDF} \cup \{l_U$ qb4o:hasAttribute $a_U\}$
      end for
      $lh_U = getPairURI(l_U, h_U)$
      $C_{RDF} \leftarrow C_{RDF} \cup \{lh_U$ a qb4o:LevelInHierarchy$\}$
      $C_{RDF} \leftarrow C_{RDF} \cup \{lh_U$ qb4o:levelComponent $l_U\}$
      $C_{RDF} \leftarrow C_{RDF} \cup \{lh_U$ qb4o:hierarchyComponent $h_U\}$
      let P be the set of parent levels of level l according to h
      for all $p \in P$ **do**
        $p_U = getURI(level, p.name)$
        $ph_U = getPairURI(p_U, h_U)$
        $step_U = getPairURI(lh_U, ph_U)$
        $C_{RDF} \leftarrow C_{RDF} \cup \{step_U$ a qb4o:HierarchyStep $\}$
        $C_{RDF} \leftarrow C_{RDF} \cup \{step_U$ qb4o:childLevel $lh_U\}$
        $C_{RDF} \leftarrow C_{RDF} \cup \{step_U$ qb4o:parentLevel $ph_U\}$
        $C_{RDF} \leftarrow C_{RDF} \cup \{step_U$ qb4o:cardinality qb4o:OneToMany $\}$
      end for
    end for
  end for
end for

for all $m \in M$ **do**
  $m_U = getURI(m.name)$
  $aggFunc_{RDF} = getAggregateFunction(m.aggregateFunction)$
  $C_{RDF} \leftarrow C_{RDF} \cup \{m_U$ a qb:MeasureProperty $\}$
  aux = [qb:measure m_U;qb:aggregateFunction $aggFunc_{RDF}$]
  $C_{RDF} \leftarrow C_{RDF} \cup \{c_U$ qb:component $aux\}$
end for

let $FactLevels$ be the set of levels that participate in cube C
for all $fl \in FactLevels$ **do**
  $fl_U = getURI(fl.name)$
  aux = [qb4o:level fl_U; qb4o:cardinality qb4o:ManyToOne]
  $C_{RDF} \leftarrow C_{RDF} \cup \{c_U$ qb:component $aux\}$
end for
let M be the set of measures in cube C

Algorithm 2 Creating a cube instance in QB4OLAP from a cube in Mondrian Schema

Input: C_M: internal representation of cube C obtained from Mondrian schema.
Output: C_{R2RML}: a set of R2RML mappings that transform relational data in cube C into RDF and QB4OLAP.

$C_{R2RML} \leftarrow \emptyset$
let D be the set of dimensions in C_M
for all $d \in D$ **do**
  let L be the set of levels in dimensions d
  for all $l \in L$ **do**
    $ltm_U = getTripleMapURI(l.name)$
    $C_{R2RML} \leftarrow C_{R2RML} \cup \{ltm_U$ a rr:TriplesMap$\}$
    $C_{R2RML} \leftarrow C_{R2RML} \cup \{ltm_U$ rr:logicalTable [rr:tableName "$l.tableName$"]$\}$
    $template = getTemplate(l.name, l.keyColumn)$
    $sm = templateTermMap(subject, template,$rr:IRI$)$
    $C_{R2RML} \leftarrow C_{R2RML} \cup \{ltm_U$ rr:subjectMap $sm\}$
    $pm = constantTermMap(predicate,$ qb4o:inLevel)
    $om = constantTermMap(object, getURI(l.name))$
    $pom = predicateObjectMap(pm, om)$
    $C_{R2RML} \leftarrow C_{R2RML} \cup \{ltm_U$ rr:predicateObjectMap $pom\}$
    let A be the set of attributes in level l
    for all $a \in A$ **do**
      $pm = constantTermMap(predicate, getURI(a.name))$
      $om = columnTermMap(object, a.columnName,$rr:Literal$)$
      $pom = predicateObjectMap(pm, om)$
      $C_{R2RML} \leftarrow C_{R2RML} \cup \{ltm_U$ rr:predicateObjectMap $pom\}$
    end for
    let P be the set of parent levels for level l
    for all $p \in P$ **do**
      $template = getTemplate(p.name, p.keyColumn)$
      $pm = constantTermMap(predicate,$ skos:broader)
      $om = templateTermMap(object, template,$ rr:IRI$)$
      $pom = predicateObjectMap(pm, om)$
      $C_{R2RML} \leftarrow C_{R2RML} \cup \{ltm_U$ rr:predicateObjectMap $pom\}$
    end for
  end for
end for

$ctm_U = getTripleMapURI(C.name)$
$template = getTemplate(c.name, c.factKey)$
$C_{R2RML} \leftarrow C_{R2RML} \cup \{ctm_U$ a rr:TriplesMap $\}$
$C_{R2RML} \leftarrow C_{R2RML} \cup \{ctm_U$ rr:logicalTable [rr:tableName "$c.factTable$"]$\}$
$sm = templateTermMap(subject, template,$ rr:IRI$)$
$C_{R2RML} \leftarrow C_{R2RML} \cup \{ctm_U$ rr:subjectMap $sm\}$
$pm = constantTermMap(predicate,$ qb:dataSet)
$om = constantTermMap(object, getURI(c.dataSet))$
$pom = predicateObjectMap(pm, om)$
$C_{R2RML} \leftarrow C_{R2RML} \cup \{ltm_U$ rr:predicateObjectMap $pom\}$
let $FactLevels$ be the set of levels that participate in cube C
for all $fl \in FactLevels$ **do**
  $template = getTemplate(fl.name, fl.keyColumn)$
  $pm = constantTermMap(predicate, getURI(fl.name))$
  $om = templateTermMap(object, template,$ rr:IRI$)$
  $pom = predicateObjectMap(pm, om)$
  $C_{R2RML} \leftarrow C_{R2RML} \cup \{ltm_U$ rr:predicateObjectMap $pom\}$
end for
let M be the set of measures in cube C
for all $m \in M$ **do**
  $pm = constantTermMap(predicate, getURI(m.name))$
  $om = columnTermMap(object, m.columnName,$rr:Literal$)$
  $pom = predicateObjectMap(pm, om)$
  $C_{R2RML} \leftarrow C_{R2RML} \cup \{ltm_U$ rr:predicateObjectMap $pom\}$
end for

Figure 4 shows the conceptual design of the cube, using the MultiDim model [12].

As presented in Section III our solution generates an RDF representation of a multidimensional data cube implemented as ROLAP, specifically implemented over Pentaho Mondrian. A ROLAP implementation of a multidimensional model consists of a set of relational tables. The *fact table* stores the measured values. Each tuple in the fact table is related to one level member in each of the dimensions that participate in the cube. Dimensions can be represented in different ways: one denormailzed table for each dimension (*star schema*), several normalized tables for each dimension (*snowflake schema*) and mixed approaches. Figure 5 presents the star schema that implements the conceptual model presented in Figure 4, while Figure 6 shows some instances for each of the tables in the star schema. Figure 6d only shows the columns in the Geography table that are involved in the level path Home→City→State.

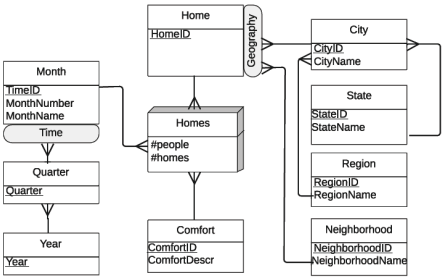


Figure 4: Conceptual multidimensional schema (Homes cube).

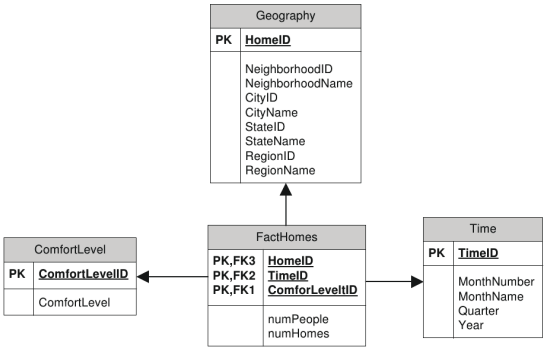


Figure 5: Logical design (Homes cube).

A. Using QB4OLAP Engine on the Homes Cube

We next show some excerpts of the result of using QB4OLAP Engine to transform the Homes cube. We begin showing the triples generated by the *cube schema translator*, followed by a sample of the R2RML mappings produced by the *cube instance translator*. We also show the triples resulting of applying these R2RML mappings to the relational representation of the cube.

1) Representation of the Cube Structure: Figure **??** shows a sample of the result of applying Algorithm 1 to the data cube presented in Figure 4. A set of prefixes (Lines 1-7) are followed by the cube structure (Lines 9-17). The rest of the triples define the Geography dimension. First, the dimension itself is defined (lines 20-22), followed by the the hierarchies in the dimension (lines 23-28). Notice that in this case there is only one hierarchy that contains all the levels. Due to space restrictions we only show the definition of levels Home and City (lines 29-35). Since levels may belong to different hierarchies, and in each hierarchy each level may have different parents, QB4OLAP represents parent-child relationships between levels as relationships of pairs (level,hierarchy). These pairs are modeled as instances of `qb4o:LevelInHierarchy`. Lines 36-42 define these instances for the levels Home and City. The relationships between pairs are represented as instances of tttqb4o:HierarchyStep, which allows to add properties to each step, such as cardinality restrictions. For example, Lines 43-47 tell that the City level is the parent of the Home level.

(a) FactHomes tableName

HomeID	TimeID	ComfortLevelID	numPeople	numHomes
2011000016	201109	1	2	1
2011000434	201103	2	1	1
2011002342	201102	3	2	1

(b) ComfortLevel table

ComfortLevelID	ComfortLevel
1	LOW
2	MEDIUM-LOW
3	MEDIUM-HIGH
4	HIGH

(c) Time table

TimeID	MonthNumber	MonthName	Quarter	Year
201102	2	February	Q12011	2011
201103	3	March	Q12011	2011
201109	9	September	Q32011	2011

(d) Geography table

HomeID	CityID	CityName	StateID	StateName
2011000016	05320	Colonia del Sacramento	5	Colonia
2011002342	10024	Punta del Este	10	Maldonado
2011000434	10024	Punta del Este	10	Maldonado

Figure 6: Sample instances of the star schema of the Homes cube

2) Representation of a Cube Instance: As presented in Section III-C, QB4OLAP Engine generates a set of R2RML mappings to transform the relational representation of the data cube into RDF. In Figure **??** we show, as an example, the R2RML mappings that populate levels Home and City in the Geography dimension. When these mappings are applied to relational data, a set of RDF triples are obtained. Figure **??** shows the result of applying the R2RML mappings from Figure **??** to the relational instances described in Figure 6.

B. Querying a Cube in QB4OLAP

QB4OLAP cubes can be analyzed *à la* OLAP using standard SPARQL queries. Due to the fact that dimension hierarchies and parent-child relationships between levels are explicitly represented using RDF triples, QB4OLAP allows to compute aggregated data from data published at a certain granularity level. In other approaches, like QB, this is not possible and data at different granularities need to be computed before publishing it as RDF. Take for instance Example 1, where the Geography dimension is traversed following the Home→City→State level path. This query corresponds to performing a rollup operation over the cube, from level Home to level State, and a slice operation to keep only the dimensions Geography and ComfortLevel. It is important to remark that the information encoded in the cube schema allows to obtain this query automatically.

```
@prefix qb: <http://purl.org/linked−data/cube#>.
@prefix qb4o: <http://purl.org/qb4olap/cubes#>.
@prefix cube: <http://www.fing.edu.uy/inco/cubes#> .
@prefix dims: <http://www.fing.edu.uy/inco/cubes/dimensions#> .
@prefix hier: <http://www.fing.edu.uy/inco/cubes/hierarchies#> .
@prefix ms: <http://www.fing.edu.uy/inco/cubes/measures#> .
@prefix att: <http://www.fing.edu.uy/inco/cubes/attributes#> .

# Cube definition (Data structure)
cube:Homes a qb:DataStructureDefinition ;
qb:component [qb4o:level dims:geography;
   qb4o:cardinality qb4o:ManyToOne] ;
qb:component [qb4o:level dims:time;
   qb4o:cardinality qb4o:ManyToOne] ;
qb:component [qb4o:level dims:comfort;
   qb4o:cardinality qb4o:ManyToOne] ;
qb:component [qb:measure ms:numPeople;
   qb4o:aggregateFunction qb4o:sum] ;
qb:component [qb:measure ms:numHomes;
   qb4o:aggregateFunction qb4o:sum] .

# Geography dimension
dims:geography a qb:DimensionProperty;
        rdfs:label "Geography dimension"@en;
        qb4o:hasHierarchy hier:geographyHier.
# Geography dimension hierarchy
hier:geographyHier a qb4o:HierarchyProperty;
        rdfs:label "Geography Hierarchy"@en;
        qb4o:inDimension dims:geography;
        qb4o:hasLevel dims:geo−home,dims:geo−neighborhood,
        dims:geo−state,dims:geo−city, dims:geo−region.
# Geography dimension levels
dims:geo−home a qb4o:LevelProperty;
        rdfs:label "Home Level"@en;
        qb4o:inHierarchy hier:geographyHier.
dims:geo−city a qb4o:LevelProperty;
        rdfs:label "City Level"@en;
        qb4o:inHierarchy hier:geographyHier.
# Organize levels in hierarchies
_:geo−home−geographyHier a qb4o:LevelInHierarchy ;
        qb4o:levelComponent dims:geo−home ;
        qb4o:hierarchyComponent hier:geographyHier.
_:geo−city−geographyHier a qb4o:LevelInHierarchy ;
        qb4o:levelComponent dims:geo−city ;
        qb4o:hierarchyComponent hier:geographyHier.
_:hs1 a qb4o:HierarchyStep;
        qb4o:childLevel _:geo−home−geographyHier ;
        qb4o:parentLevel _:geo−city−geographyHier ;
        qb4o:cardinality qb4o:OneToMany.
```

(a) Sample of the Homes cube schema

```
<#geography−HomesMap> a rr:TriplesMap;
rr:logicalTable [ rr:tableName "Geography" ];
rr:subjectMap [rr:termType rr:IRI ;
 rr:template "http://www.fing.edu.uy/inco/cubes/dic/geoHome#{HomeID}"];
rr:predicateObjectMap [ rr:predicate qb4o:inLevel;
 rr:object dims:geo−home];
rr:predicateObjectMap [ rr:predicate skos:broader;
 rr:objectMap [ rr:termType rr:IRI ;
  rr:template "http://www.fing.edu.uy/inco/cubes/dic/geoCity#{CityID}"];].
```

(b) Sample of the R2RML mappings that populate the Geography dimension

```
@prefix ns1:<http://www.fing.edu.uy/inco/cubes/dic/geoHome#>.
@prefix ns2:<http://www.fing.edu.uy/inco/cubes/dic/geoCity#>.
@prefix ns3:<http://www.fing.edu.uy/inco/cubes/dic/geoState#>.
@prefix dims:<http://www.fing.edu.uy/inco/cubes/dimensions#>.

ns1:2011000016 qb4o:inLevel dims:geo−home;
   skos:broader ns2:05320.
ns2:05320 qb4o:inLevel dims:geo−city;
   rdfs:label "Colonia del Sacramento";
   skos:broader ns3:10.
```

(c) Sample of the RDF triples obtained applying R2RML mappings

Figure 7: Sample RDF triples

Example 1. *Total number of homes and people surveyed, by state and comfort level.*

```
SELECT ?stateName ?comfLevel
  (sum(?numHomes) as ?totalHomes) (sum(?numPeople) as ?totalPeople)
WHERE { ?o a qb:Observation; dims:geo−home ?home;
  dims:comfort−comfort ?comfort ; ms:numPeople ?numPeople;
  ms:numHomes ?numHomes.
  ?home skos:broader ?city. ?city qb4o:inLevel dims:geo−city .
  ?city skos:broader ?state. ?state qb4o:inLevel dims:geo−state .
  ?state rdfs:label ?stateName .?comfort atts:comfortLevel ?comfLevel }
GROUP BY ?stateName ?comf ?comfName
```

Example 2 also performs an aggregation over the Geography dimension, and shows how easy is to implement a Top-k query using SPARQL `ORDER BY` and `LIMIT` clauses.

Example 2. *Top 10 cities with most surveyed homes.*

```
SELECT ?cityName (SUM(?numHomes) AS ?totalHomes)
WHERE {
  ?o a qb:Observation; dims:geo−home ?home; ms:numHomes ?numHomes.
  ?home skos:broader ?city. ?city qb4o:inLevel dims:geo−city .
  ?city atts:cityName ?cityName }
GROUP BY ?cityName
ORDER BY DESC (?totalHomes) LIMIT 10
```

Example 3 combines aggregation and Top-k with filtering. A BGP is used to keep only the facts (observations) that refer to LOW comfort level homes (line 7).

Example 3. *Top 3 states with the largest number of LOW comfort level homes.*

```
SELECT ?stateName (SUM(?lowHomes) AS ?lowComHomes)
WHERE {
  ?o a qb:Observation; dims:geo−home ?home; ms:numHomes ?lowHomes.
  ?home skos:broader ?city. ?city qb4o:inLevel dims:geo−city .
  ?city skos:broader ?state. ?state qb4o:inLevel dims:geo−state .
  ?state atts:stateName ?stateName.
  ?o dims:comfort−comfort
     <http://www.fing.edu.uy/inco/cubes/dic/comfort#LOW> }
GROUP BY ?stateName
ORDER BY DESC (?lowComHomes) LIMIT 3
```

Example 4 compares aggregated values that correspond to different periods of time. SPARQL `FILTER` clause is used to state the comparison criteria.

Example 4. *Total number of surveyed homes by comfort level compared to those of the previous month.*

```
SELECT ?comfLevel ?yearNo ?monthNo ?tHomes ?tHomes1
WHERE {
{ SELECT ?comfort ?yearNo ?monthNo (SUM(?numHomes) AS ?tHomes)
  WHERE { ?o a qb:Observation; dims:geo−home ?home;
   dims:comfort−comfort ?comfort ; dims:time−month ?month;
   ms:numHomes ?numHomes. ?comfort atts:comfortName ?comfName .
   ?month qb4o:inLevel dims:time−month. ?month skos:broader ?quarter.
   ?month atts:monthNumber ?monthNo .
   ?comfort atts:comfortLevel ?comfLevel.
   ?quarter qb4o:inLevel dims:time−quarter.?quarter skos:broader ?year.
   ?year qb4o:inLevel dims:time−year . ?year atts:year ?yearNo.}
  GROUP BY ?comfLevel ?yearNo ?monthNo }
  OPTIONAL {
  {SELECT ?comfort ?yearNo1 ?monthNo1 (SUM(?numHomes) AS ?tHomes1)
   WHERE { ?o a qb:Observation; dims:geo−home ?home;
    dims:comfort−comfort ?comfort ; dims:time−month ?month1;
    ms:numHomes ?numHomes. ?comfort atts:comfortName ?comfName .
    ?month1 qb4o:inLevel dims:time−month. ?month1 skos:broader ?quarter1.
    ?month1 atts:monthNumber ?monthNo1 .
    ?comfort atts:comfortLevel ?comfLevel.
    ?quarter1 qb4o:inLevel dims:time−quarter.?quarter1 skos:broader ?year1.
```

```
    ?year1 qb4o:inLevel dims:time−year . ?year1 atts:year ?yearNo1.}
   GROUP BY ?comfLevel ?yearNo1 ?monthNo1 } }
FILTER(
( (?monthNo = ?monthNo1 + 1) && (?yearNo = ?yearNo1) ) ||
( (?monthNo = 1) && (?monthNo1 = 12) && (?yearNo = ?yearNo1+1) ) ) }
GROUP BY ?comfLevel ?yearNo ?monthNo
ORDER BY ?comfLevel ?yearNo ?monthNo
```

V. Related Work

Several organizations publish statistical data using the RDF Data Cube (QB) vocabulary[14], although few information is available on the tools and processes used to build these data cubes. Some tools have been developed in the context of the Eurostat-Linked Data project[15], but they take as input statistical data in SDMX format.

There exist tools for analyzing and querying data cubes in QB. CubeViz[16] is a faceted browser on QB data cubes. OLAP4LD [13] proposes to implement OLAP operations as SPARQL queries over cubes expressed in QB. However, hierarchies and levels must be added to the cubes in order to implement the operations that need these concepts. The authors analyze the performance of their proposal, and suggest using aggregate views to materialize aggregations and speed-up queries.

VI. Conclusion and Future Work

In this paper we have presented QB4OLAP Engine, a prototype of a tool that allows to transform multidimensional ROLAP data into RDF, using QB4OLAP. This prototype shows the feasibility of our approach, while further testing of the tool is needed to reveal the scalability and performance characteristics of our implementation. Future work regarding QB4OLAP Engine includes allowing the reuse of already generated QB4OLAP data, and studying how to deal with changes in the underlying relational data. We also plan on studying mechanisms to obtain QB4OLAP data cubes using other data sources, not necessarily multidimensional data sources.

References

[1] J. Cohen, B. Dolan, M. Dunlap, J. Hellerstein, and C. Welton, "MAD Skills: New analysis practices for big data," *Proceedings of the VLDB Endowment*, vol. 2, no. 2, pp. 1481–1492, 2009.

[2] M. Golfarelli, "Open source BI platforms: A functional and architectural comparison," in *Data Warehousing and Knowledge Discovery*, ser. LNCS. Springer, 2009, vol. 5691, pp. 287–297.

[3] T. Heath and C. Bizer, *Linked Data: Evolving the Web into a Global Data Space*, ser. Synthesis Lectures on the Semantic Web. Morgan & Claypool Publishers, 2011.

[14] http://www.w3.org/2011/gld/wiki/Data_Cube_Implementations

[15] http://eurostat.linked-statistics.org/

[16] http://aksw.org/Projects/CubeViz.html

[4] L. Etcheverry and A. Vaisman, "QB4OLAP: A vocabulary for OLAP cubes on the semantic web," in *Proc. of the 3rd. International Workshop on Consuming Linked Data, COLD 2012*. Boston, USA: CEUR-WS.org, 2012.

[5] L. Etcheverry and A. A. Vaisman, "Enhancing OLAP analysis with web cubes," in *Proc. of the 9th Extended Semantic Web Conference, ESWC 2012*, ser. LNCS 7295, Crete, Greece, 2012, pp. 469–483.

[6] G. Klyne, J. J. Carroll, and B. McBride, "Resource Description Framework (RDF): Concepts and Abstract Syntax," 2004. [Online]. Available: http://www.w3.org/TR/rdf-concepts/

[7] D. Brickley, R. Guha, and B. McBride, "RDF Vocabulary Description Language 1.0: RDF Schema," 2004. [Online]. Available: http://www.w3.org/TR/rdf-schema/

[8] D. Beckett and T. Berners-Lee, "Turtle - Terse RDF Triple Language," 2011. [Online]. Available: http://www.w3.org/TeamSubmission/turtle/

[9] E. Prud'hommeaux and A. Seaborne, "SPARQL 1.1 Query Language for RDF," 2011. [Online]. Available: http://www.w3.org/TR/sparql11-query/

[10] S. Das, S.Sundara, and R. Cyganiak, "R2RML: RDB to RDF Mapping Language," 2012. [Online]. Available: http://www.w3.org/TR/r2rml/

[11] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data-the story so far," *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 5, pp. 1–22, 2009.

[12] A. Vaisman and E. Zimányi, *Data Warehouse Systems: Design and Implementation*. Springer, 2014.

[13] B. Kämpgen and A. Harth, "No size fits all - running the star schema benchmark with SPARQL and RDF aggregate views," in *The Semantic Web: Semantics and Big Data*, ser. LNCS. Springer, 2013, vol. 7882, pp. 290–304.

Inside-In Search: an alternative for performing ancillary search tasks on the Web

Ricardo Cava, Carla M. D. S. Freitas
Instituto de Informática
Universidade Federal do Rio Grande do Sul (UFRGS)
Porto Alegre – RS, Brazil
{racava,carla}@inf.ufrgs.br

Eric Barboni, Philippe Palanque, Marco Winckler
ICS-IRIT
Université Paul Sabatier
Toulouse, France
{barboni, palanque, winckler}@irit.fr

***Abstract*— Some of the search tasks users perform on the Web aim at complementing the information they are currently reading in a Web page: they are ancillary search tasks. Currently, the standard way to support such ancillary searches follows an inside-out approach, which means that query results are shown in a new window/tab or as a replacement of the current page. We claim that such inside-out approach is only suitable if users really want to dissociate the search results from the Web page they were reading. In this paper we propose an alternative approach, called "inside-in", where query results are displayed inside the Web page next to the keyword that motivated the user to launch an ancillary search. In order to demonstrate the feasibility of our approach we have developed a tool that embeds an egocentric information visualization technique in the Web page. This tool supports nested queries and allows the display of multiple data attributes. The approach is illustrated by a case study based on ancillary searches of coauthors from a digital library. The paper also reports some preliminary results obtained with an experiment conducted with remote users.**

Keywords*— *Visualization techniques; Web interaction techniques; information retrieval; ancillary Web queries; Web-based user interfaces.

I. INTRODUCTION

Search tasks might vary in complexity: they can be relatively simple such as looking for current weather forecast or otherwise tricky and time consuming such as collecting all information required for planning a vacation trip overseas. Over the years, users have become used to retrieve information from the Web, and for that they developed several strategies, which can be summarized as information lookup and exploratory search [16]. Whilst exploratory search requires time for scanning and reading documents, information lookup can be solved by simple factual question-answer interactions.

Regardless the users' need for information, the user interface provided by such information retrieval systems must be simple enough to allow users to formulate queries and understand the results provided by search engines [10]. Quite often, queries start by filling a search box with keywords. Formulating queries in this way is a daunting task that requires users to open a new window, to fill in a form with appropriate keywords, and then scan the list of results until finding the one that corresponds to the goal. Nonetheless, many search tasks can also be accomplished by browsing (or navigating) among interlinked documents. Browsing is often preferred by users because it is cognitively less demanding to recognize a keyword than typing one, but the advantages of browsing is quickly lost if users have to visit many links until reaching the desired information [15].

Currently, the standard way to show search results (obtained by either filling in a form or browsing documents) follows an *inside-out search* approach, which means, results are displayed in a new window/tab and/or replace the current window/tab's contents. We assume that such *inside-out search* approach is only suitable when users want to dissociate the search results from the Web page they are reading. However, some searches that users perform on the web are done just to complement information they are currently reading [12], the so-called *ancillary searches*. For example, users reading an article in a Web page might be curious to know with whom the author of that particular article has published in the past. In this scenario, looking up for co-authors constitutes an ancillary search, which is not meant to divert users' attention from reading the article. For such kind of ancillary-search tasks, using the *inside-out* approach to display results in a new window/tab might be unsuitable since it creates an articulatory distance between the origin of the request and the information display making difficult to users to assess if their goal has been fulfilled or not by the query.

In this paper we propose an alternative approach, called *inside-in* search, where ancillary searches are launched by users from inside the Web page they are currently reading, and the corresponding results are displayed as a contextual help next to the keywords used to formulate query. Giving that the amount of search results from an ancillary search can be huge, we assume that users would benefit from an interactive information visualization technique that provides nested queries (ex. co-authors of a particular co-author), and allows the display of multiple attributes as ancillary data (ex. co-authors, number of joint publications, type of publications in common).

In order to demonstrate the feasibility of the *inside-in* approach, we have developed a supporting tool that embeds an egocentric information visualization technique called IRIS. For the purpose of this paper we present a case study focused on the visualization of co-authors extracted from the DBLP. We present the theoretical background (section II), followed by a task analysis of ancillary searches using an *inside-in* approach (section III). Sections IV and V present

978-1-4799-6953-1/14 $31.00 © 2014 IEEE
DOI 10.1109/LAWeb.2014.21

the implementation of our approach and some preliminary results obtained from a remote user testing experiment. Section VI briefly reviews related work, and section VII concludes our paper, emphasizing the current contribution and providing some insights on future work.

II. THEORETICAL BACKGROUND

In order to better understand the tasks accomplished by users while performing search tasks, Sutcliffe and Ennis [23] propose an information seeking process that encompasses the following steps: i) problem identification, ii) articulation of information need(s), iii) query formulation, and iv) results evaluation. These steps recall Norman's cognitive model [19], which explains the gulfs that exist in the communication between users and systems: the *execution gulf* and the *evaluation gulf*. These concepts were introduced by D. Norman (1986) and became popular through the book [18].

The *execution gulf* is the effort required for a user to express an intention in terms of commands or instructions. In other words, the *gulf of execution* is the difference between the intentions of the users and what the system allows them to do or how well the system supports those actions. For example, if the users find an unknown word whilst navigating the web, they might expect that clicking on a link (on that word) would provide them with the complimentary information required to understand the meaning of the word. In the user's language "click the link" defines the goal for obtaining the word's meaning. However, if the link does not provide the expected results, users have to execute additional actions, such as opening a new window, visiting a search web site, typing the adequate keywords to specify the search, and, finally, browsing the list of results until getting the desired definition.

The *evaluation gulf* refers to the way the results provided by the system are meaningful or understandable by the users, and in accordance with their goals. In other words, the *gulf of evaluation* is the degree to which the system or artifact provides representations that can be directly perceived and interpreted in terms of the user's expectations and intentions. Thus, if the system does not "present itself" in a way that lets the users derive which sequence of actions will lead to the intended goal or system state, or infer whether previous actions have moved them closer to their goal, there is a large *gulf of evaluation*. In our case, the users must spend a considerable amount of effort and significant attentional resources to perform a query in a new window, to identify the answers that correspond to their expectations and, then, to put the word's meaning back in the appropriate context.

Overall, the *gulfs of evaluation and of execution* refer to the mismatch between our internal goals on the one side, and, on the other side, the expectations and the availability of information specifying the state of the world (or an artifact) and how we may change it [18].

Fig. 1 presents a revised version of the execution and evaluation gulfs cognitive model [19] explicitly showing the *articulatory distance* and the *semantic distance* both in terms of user *input* (i.e. when users formulate the query) and in terms of system *output* (i.e. when the system displays the results to be assessed by the user).

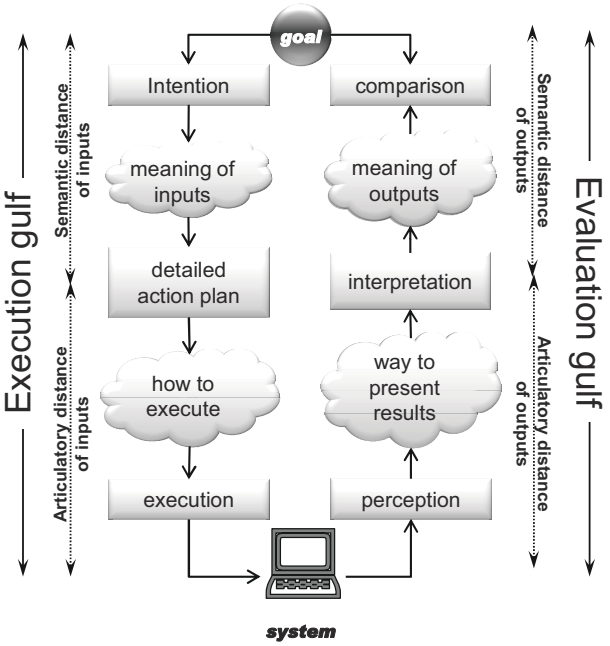


Figure 1. Execution and evaluation gulfs in search tasks, adapted from the Norman's model [19].

Based on this model, we can compare two user interfaces for looking for information on the Web: *browsing* and *filling-in forms*. The *semantic distance of input* can be considered smaller in *browsing* than *filling-in forms* because recognition of words is less cognitive demanding than choosing the appropriate keywords. The *articulatory distance of input* is also smaller for *browsing* because it just requires a click, which is much faster than typing words. However, the *articulatory distance of output* is bigger when users have to navigate/*browse* many Web pages to find the information. This *articulatory distance* is smaller when the list of pages possibly containing the information needed is displayed in a rank. Nonetheless, ranks might increase the *semantic distance of output* as users might not (necessarily) understand how the ranking was created and in which extension the search results are relevant to the information s/he looking for.

III. TASK ANALYSIS OF ANCILLARY-SEARCH TASKS

Some of the difficulties users have to face when using an *inside-out* approach become evident when users perform ancillary search tasks. To illustrate this, we present below a scenario, which ultimately require users to perform ancillary searches for better understanding the contents of an article s/he reading on the Web.

A. Motivating scenario

In order to ground the scenarios around the same application domain, we have chosen to illustrate them with data about co-authorship, as follows:

"John has been recently appointed as expert member of the jury that will evaluate the research of a Graduate Program in Computer Science at an university in the South of France. John received a Web form which contains the list of ~400 researchers for which he has to provide an assessment based on the number of co-authors and relevant publications they have in the field. The number of publications and co-authors is required to calculate two important metrics: the researcher's productivity (accordingly to a formula that takes into account the number of co-authors to estimate the individual effort for the publication) and the size of collaboration network (considered that successful scientific collaborations ultimately lead to joint publications). So, John starts by making a Google search on the Web using the name of the first researcher in the list. Find the right researcher's Web page was not easy as the Google search engine returns many entry points including homonymous and some trash pages. After fixing typos and refining the terms of the query, John found the researcher's Web page where he can count the number of his publications; the mental calculation to accomplish this task is easy. Now, for assessing the size of the research network, things are more complicate. John considers two options: i) to create manually a side-list with the names of co-authors; or ii) look for them in the DBLP web site. John chose the second option so he types the name of the researcher on the search box of the navigation, goes to DBLP web site, scrolls down to reach the zone where co-authors are displayed, and open up the list of co-authors. Now John is ready to fill in the form but then he realizes that the DBLP contents now occupy the window that previously contained the Web form … For the next 399 researchers John decided to create new tabs for keeping the DBLP search apart from the Web form. Then, he found out himself being performing repetitive copy-and-paste between tabs which did not improve his overall performance…"

As we can observe in the described scenario, searching co-authors should be considered an ancillary search that complements the user's main task, which is filling in the Web form. From this scenario we find some issues that make the following users tasks difficult:

- Formulating queries is error-prone (might contain typos) and also time consuming (typing takes time).
- Keywords might be ambiguous and generic search engines will return broad results. Users may have to scan the list of results until finding the one that corresponds to their goals.
- There are many alternative locations for showing results (including new windows/tabs); choosing the best location for displaying results depends on where the results are meant to be used.
- Some queries might be repetitive; so, saving a few seconds in the time required to complete a single task might represent a huge improvement in the overall task performance at the end of the day.

B. *Rational for improving ancillary-search tasks*

We claim that the issues raised above can be solved (or at least minimized) with an *inside-in* approach including the following mechanisms aimed at supporting ancillary-search tasks:

- Launching queries from words available in the current Web page can reduce typos. Keywords can be selected with mouse click, which is sensibly faster than typing in using a keyboard.
- Ambiguous results are often the result of a broad search. This problem can be reduced by providing specialized queries that operate on specific application domains using user-selected keywords.
- Query results can be shown inside the current page inline to the selected keywords. This is one of the keystones for the *inside-in* approach, but notice that queries should be launched on user's demand. If the system systematically launches queries without user's request, the Web page will become polluted by ancillary results, and the benefits of the *inside-in* approach will be lost.
- Ancillary results should support some kind of interaction to allow users to perform nested queries. This element is important for repetitive tasks, which are often associated to contexts where ancillary searches are required.

The selection of keywords on the text and the use of predefined queries aim at reducing the gulf of execution. This reduction is achieved by minimizing the users' effort in informing keywords to the system (articulatory distance of inputs) and by favoring recognition of keywords and queries rather than imposing the formulation of complete queries (semantic distance of inputs). Predefined queries also help to reduce the evaluation gulf as the results are focused on a particular application domain (semantic distance of outputs). By showing results in the same page and allowing the user to perform nested queries, the *inside-in* approach helps to reduce the *articulatory* distance of outputs.

IV. IMPLEMENTATION OF THE *INSIDE IN* APPROACH

In order to support the proposed *inside-in* approach, we have developed a framework whose main principles are briefly illustrated in Fig. 2. Our *inside-in* approach is built upon the concept of Web augmentation [3], which defines strategies for implementing tools that can extend the set of elementary tasks users can do while navigating on the Web. Whilst the full details about the implementation of that framework are out of the scope of this paper, it is interesting to notice that it includes a client-side module and a broker at the server-side.

A. *The server side module*

The "broker" at the server side (Fig. 2.b) contains a set of preprogramed query functions that are made available to the end users. The query broker was originally conceived to support robust integration of many similarity functions using approximated data instance matching [8]. Nonetheless, the framework is extensible and can accommodate new

specialized queries. The number of queries accessible from the client-side can be configured dynamically, but for the purposes of this paper we are just using a specific one, which returns the co-authors of a given researcher. The only thing users have to do is to select keywords from the set of words displayed in the current Web page, and then trigger (by a simple click) the ancillary-search query for co-authors.

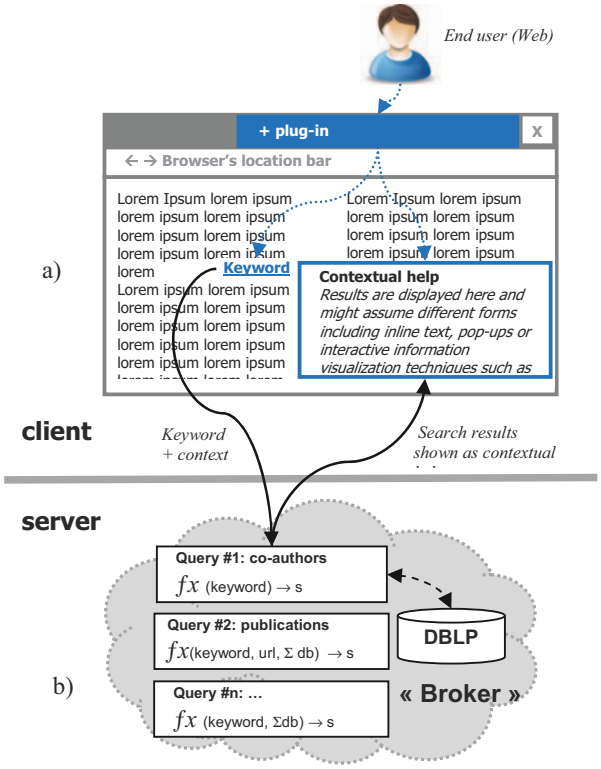


Figure 2. Overview of the framework architecture.

B. The client-side module

This module (Fig. 2.a) allows users to select keywords in the current web page and to trigger the queries for ancillary search available in the broker. Once the broker replies, this module modifies the current Web page to display the search results as a kind of contextual help. For that, the DOM structure of the current Web page is adapted using a set of JavaScript functions, called augmenters [10]. As demonstrated in a previous work [28], adaptations created by augmenters are volatile and do not affect the application running in the Web server. The client-side module can display ancillary data using different interaction techniques, including the information visualization technique IRIS, which is described in the next section.

C. The information visualization technique IRIS

The information visualization technique IRIS (which stands for Investigating Relationships between Indexes of Similarity) was originally conceived to provide visualizations of search results containing some measure of the similarity between the retrieved item and the query terms. Nonetheless, IRIS is a generic visualization technique, and can be used to display many different types of data. The inner data structure supported by IRIS is composed of a set of attributes that can be attached to two main *nodes* and *edges* featuring a radial layout [9]. One of the nodes is defined as the *centroid* to which all the other nodes are linked. Whilst *nodes* and *edges* are fixed and mandatory elements in the visualization, the list of attributes attached to them is arbitrary. Thus, IRIS can be configured to work with a large variety of data sets that could be organized around an egocentric network topology.

In Fig. 3, we illustrate the use of IRIS presenting relationships between an author and his/her *n* co-authors. The name of the author used as keyword to launch the ancillary search about co-authors is placed at the center of a circle formed by bar graphs. The size of the each bar graph represents the number of co-authored papers between the author and each co-author. The name of the co-author and an ordinal number are also displayed radially. To deal with authors that have a large number of co-authors that would not fit into the representation, we adopted a focus+context approach [5]. The focus is the darker center-right area of the circle, and the context is represented by the two sub-areas in light gray where the font is displayed in decreasing size towards the left side of the circle. This effect is obtained by a fisheye distortion [1]. As we can see, only 47 of 83 co-authors are shown. The ordinal number displayed along the name of the co-author provides information about which subset is being displayed at each moment. Users can move the position of any co-author to the focus area by clicking on his/her name.

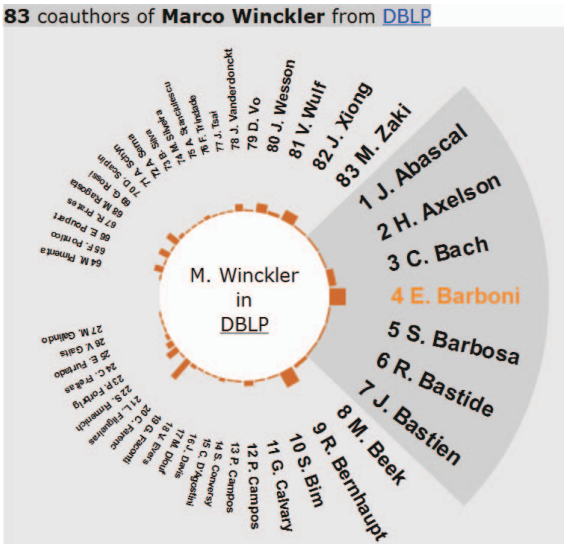


Figure 3. IRIS visualization of co-authorship of information from DBLP.

The implementation of IRIS that is delivered with our framework also embeds several functions for allowing interactive exploration of the dataset. In our case study of co-authorship network, a centroid is made up from the keyword selected by the users on the Web page. The nodes and

attributes are obtained as the result of an ancillary-search query.

Other details regarding the attributes can be obtained by activating tooltips which show the complete name of a selected co-author (Fig. 4.a) or the number of papers per category, as provided by DBLP (Fig. 4.b). The link shown below the name of the author gives access to the information in DBLP.

Figure 4. Details of IRIS representation showing tooltips for: a) co-author's name; and b) number of papers per category.

IRIS also provides some interaction beyond the display of tooltips. The user can place a co-author in focus by clicking on it with the left mouse button. This causes the name change its color to orange and move to the center of the focus area. The whole representation is rotated, the direction of the rotation being determined in a way to minimize the angle to be used. The movement is animated to minimize user's disorientation [13][25].

V. Evaluation

In this section we describe a remote user study we have run in order to collect real users' feedback about our tools.

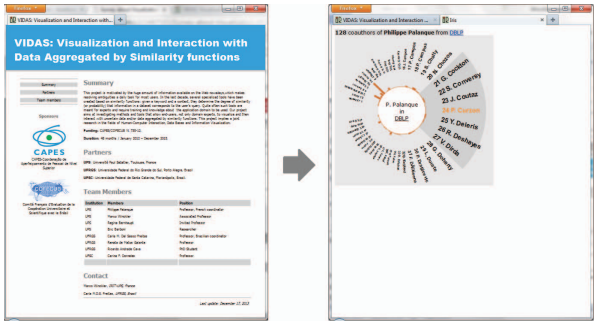
A. Rationale and scenario used in the evaluation

The study was focused on a population of users that might have the need (or curiosity) to look up certain type of information while browsing the Web. Moreover, it was important to envisage a scenario where the information needed is easily available by other means than our visualization technique, so that users could compare the different design alternatives. Therefore, we have considered a scenario where there is list of researchers shown in a web page and a user wants to check their co-authors in some bibliographic database, in the present case, DBLP, which is easily accessed at http://www.informatik.uni-trier.de/~ley/db/. So, users could use either DBLP or IRIS to see the same results.

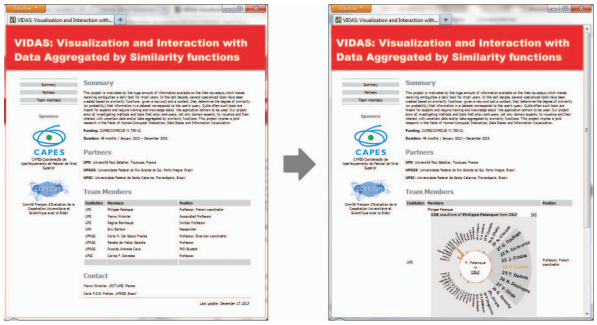
In our scenario, we assume that, given a community of end users, empirical studies can be performed to identify suggestions for ancillary questions that fulfill users' needs in a particular application domain. For the purpose of this paper, we assume that end users are researchers, principal investigators searching for postdocs, deans, experts of funding agencies, etc., for which looking up for co-authors is part of some of their tasks.

For reducing the semantic and articulatory distance of inputs, we devised a browsing mechanism that is based on clicking on names of researchers already available in a Web page. Selecting a name embedded into a Web page launches a predefined search for his/her co-authors so that users do not need to type keywords (as in DBLP or Google). Moreover, the scope of the search is predefined (i.e. co-authors) as well the data source (i.e. DBLP).

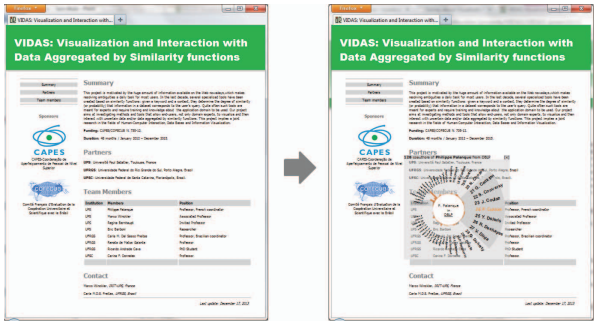
For the purpose of the survey, we used the team members of the VIDAS project's Web page as shown in Fig. 5. However, the design solutions presented herein are generic, and could be used in any Web page using the names it might contain. The list of co-authors is obtained by parsing data from DBLP and displaying it using IRIS. To see the co-authors of a team member one just need to click on (or select) his/her name in the Web page.



a) Results are shown in a new tab on the Web browser.



b) Results are embedded into the Web page changing layout.



c) Results are shown as a semi-transparent layer.

Figure 5. Alternative design options for positioning the search results.

As for the semantic and articulatory distances of outputs, our alternative designs show the search results encoded in an abstract visualization and display them in three ways as illustrated in Fig. 5:

1. *Displayed in another tab/window*. We assume that the articulatory distance of output is greater in this option than in the other ones as results are not shown in the same Web page where the search was triggered.

2. *Embedded into the current pag*e. Notice that this option will change the Web page layout on the client-side to place IRIS next to the position selected by the user. However, users can easily go back to the original page layout just by clicking on the icon [x] available at the top-right site of IRIS panel. Articulatory distance of output is reduced by placing results in the same page, though changing Web layout could be disturbing for some users.

3. *Placing IRIS as a semi-transparent layer over the page*. In this case, the Web page layout does not change, but IRIS can hide some information. Nonetheless, users can freely move IRIS around.

B. Procedures

An online survey was created using the tool SurveyGizmo. The survey was organized in five main chapters including:

- *Presentation of the study:* this step described the task "search co-authors" and how to launch IRIS.
- *About ancillary queries:* using a Likert scale participants were asked to tell if they think whether (or not) "*starting a query directly from a Web page*" by clicking on researchers names was useful, improved performance, and how much they liked it. They were also asked to tell what were the most frequent ancillary queries they triggered while navigating the Web.
- *About IRIS:* using a Likert scale participants were asked if they think that "*showing results of queries on the Web by means of an interactive visualization technique*" was useful, improved performance, and how much they liked it. They were also asked to tell if they prefer a flat HTML page to show query results or IRIS.
- *About alternative locations for displaying search results:* participants were asked to tell which they prefer: ancillary query results in a new tab/window, results embedded into the Web page, the option of results floating over the web page, or none of these.
- *About participants:* the last step addressed the profile of participants including demographic data.

C. Participants

Participants were recruited via mailing lists and social networks of authors involved in this work. The survey was delivered in English regardless the origin of the participants. We have addressed the survey to professors, researchers and students involved in research activities, as that population might feel concerned by the underlying task of looking up information about co-authors as part of their activities. In the short period of 3 weeks, while the survey remained open, we received 150 visitors but only 61 complete responses were used to validate our hypotheses, as described below.

Most of participants were male (77.1%) and, in average 26.1 years old (SD=5.7), being 55.7% between 25-34 years old, 24.6% between 35-54 years old and 19.7% between 18-24 years old. Among the participants 44.3% were students, 39.3% researchers/professors and 16.4% work in the industry. The highest degree included 26.2% of PhD, 44.3% M.Sc. degree, 19.7% Bachelor's degree and 9.8% are still undergraduate students. We have got responses from Argentina, Austria, Brazil, France and Spain. Participants estimated to spend ~5.3 hours (SD=4.1) per week using search engines over the Web, most of which (~4.3 hours per week, SD=3.9) is spent looking for authors and publications. The amount of time participants perform searches related to authors and publications qualify them as typical users of tools with predefined queries for looking for information, such as searching for co-authors. The most frequently reported digital library was GoogleScholar with 88.1%, followed by IEEE Xplore (76.3%), SpringerLink (52.5%), ACM DL (69.5%) and DBLP (35.6%).

D. Results and discussion

The results collected with the survey are quite positive with respect to the *inside-in* approach as implemented by our tools. The results confirm that typing text to formulate a query is less appreciated than selecting keywords, as shown in Table I. Most participants found that selecting a term in a web page for launching a query is useful (N=52; 85.3%) and that it improves performance (N=49; 80,3%). Moreover, only 11 participants (18.1%) prefer typing a text to formulate queries. As mentioned by participant P23, launching a search by selecting keywords on Web pages depends on "*…the amount of text and how it is showed/arranged. The UX won't be the same with large paragraphs..."* Moreover, 47 participants (77%) provided a large list of suggestions for specialized queries that could be launched by simply selecting a term in a Web page. Considering that all open questions in the survey were optional, we might consider that the experience might have somewhat stimulated participants to contribute to the research.

TABLE I. USERS PREFERENCES FOR LAUNCHING QUERIES

	Selecting keywords in the web page						***Typing text***	
	Useful		*Improve performance*		*Prefer*		*Prefer*	
Strong agree	25	**N=52**	18	**N=49**	20	**N=49**	4	**N=11**
Agree	27	***85,3%***	31	***80,3%***	29	***80,3%***	7	***18,1%***
Neutral	1	**N=1** ***1,6%***	6	**N=6** ***9,8%***	6	**N=6** ***9,8%***	22	**N=22** ***36,1%***
Disagree	6	**N=8**	6	**N=6**	4	**N=6**	20	**N=28**
Strongly disagree	2	***13,1%***	0	***9,8%***	2	***9,6%***	8	***45,9%***

Regarding the use of a visualization technique as an alternative solution to show the results of ancillary queries, 51 participants (83.6%) said that using IRIS was useful while 44 of them (72.2%) think that it improved performance during ancillary queries. Most of participants (N=50, 82%) prefer to see the results with IRIS. These results, shown in Table II, yet preliminary, validate IRIS as a suitable design solution for the visualization of ancillary queries, at least as

far as searching for co-authors are a concern. A significant number of subjects (N=13, 21.3%) are neutral regarding improvement of performance. This can be exemplified by some comments: some users stated that it depends on the application; others posed this dependency on what is provided by the visualization technique – if the technique allows deriving information more easily than a textual list. For example, participant P34 informed to prefer the flat version because "*I am used to HTML pages so I am not sure it would improve my performance*", and participant P111 wrote: "*Only if both solutions allow me to quickly achieve my goal*". These comments suggest that improving user performance is a key factor leading to the adoption of information visualization techniques to display search results.

TABLE II. USERS PREFERENCES FOR THE DISPLAY OF SEARCH RESULTS.

	Search results shown with a visualization technique						***Textual list***	
	Useful		*Improve performance*		*Prefer*		*Prefer*	
Strong agree	23	**N=51**	12	**N=44**	20	**N=50**	1	**N=7**
Agree	28	***83,6%***	32	***72,2%***	30	***82%***	6	***11,5%***
Neutral	8	**N=8** ***13,1%***	13	**N=13** ***21,3%***	9	**N=9** ***14,8%***	16	**N=16** ***26,2%***
Disagree	2	**N=2**	4	**N=4**	1	**N=2**	27	**N=38**
Strongly disagree	0	***3,3%***	0	***6,6%***	1	***3,2%***	11	***62,3%***

Considering the alternatives for the location of IRIS, 36 participants (*59%*) said to prefer the option embedding results into the current page while 22 participants (*36,1%*) liked more the design option showing the results floating over the Web page. Most participants prefer to see ancillary results in the same page (total N=58; *95,1%*). Only 3 participants prefer to see ancillary results in a new tab/window (4,9%) as shown in Table III. Interesting enough, among the participants that prefer the floating option, some of them asked to replace transparency by an opaque background, as mentioned by participant P33: "*I like it [floating version] but it would be better if the background wasn't transparent…".*

TABLE III. USERS PREFERENCES FOR THE LOCATION OF RESULTS.

	Preferred location for IRIS
Embedded in the same page	**N=36** | ***59,0%***
Floating over the same page	**N=22** | ***36,1%***
In a new web page	**N=3** | ***4,9%***

It is also interesting to notice that most of the participants clearly pointed out that option Green (floating over the same page) and option Red (embedded in the same page) presented the advantage of reducing the interruption created by search engines when showing the results in a new window/tab. This can be illustrated by participant P67 who wrote: "*Advantage: doesn't disrupt the current page Disadvantage: requires switching tabs*". The frequency on which such disruption was reported in the comments lets us think that participants really notice the articulatory distance created when new windows are open. Moreover, this may also mean a perception of reduced performance as indicated by the comments of participants P2: "*Changing tabs (and losing my thoughts)*" and P33: "*Change of context is annoying.*"

Overall, users did not like the option that shows a new page because of the change of context. This result is compatible with the issues brought by *inside-out* search approaches. However, the few who liked that mentioned the possibility of having more space to display more information. The majority of positive comments were centered on the availability of the additional information right next to the search keyword.

VI. RELATED WORK

Search engines have become an integral part of our daily lives [12], but many users are still struggling to use them to obtain the results they need [11]. Some of the problems users have to face are related to the fact that, given the increasing availability of data in the Web, users should be very precise in the way they formulate their queries. For that, the design of search user interfaces has developed dramatically along the years, from simple keyword search systems to complex combinations of faceted filters and sorting mechanisms.

Wilson [27] claims that the design of the user interface has also an important cognitive impact on tasks performance; thus, search engines should evolve to take into account users' needs. Although these claims are valid, most of research efforts in the area have been focused on two main areas: algorithms for improving the accuracy of search engines with respect to many users concerns and approaches for improving the visualization of Web pages [24]. For example, Schwarz and Morris [21] describe an information visualization approach for augmenting Web pages with data about the credibility of the ranking proposed by the search engine. While such approach does not help users to formulate better queries, it might help users to better select the pages to visit according to the rank of pages proposed by search engines. Capra et al. [4] also proposed to augment the user interface of search results by adding images next to the ranking provided by search engines aiming at helping users to make better decisions. These few examples are illustrative of strategies for improving the design and display of the ranking of results from search engines.

In the last decades, several information visualization approaches have been developed for presenting search results coming either from search engines or widely used databases, such as DBLP, ACM DL, IEEE Xplore, etc. Some search engines with built-in visualization tools have also been developed. The first reports presenting and/or discussing visualization of search results date from the late 90's and early 2000's. Visualization approaches range from 2D plots [14][17], glyph-based techniques [6][20] to 3D designs [2][7][26]. However, although along the years, many different techniques have been evaluated [22] with results favoring visualizations, the majority of web search engines

still provide textual lists ordered by some user or tool specified criteria.

It is interesting to notice that current research efforts follow an *inside-out* approach. In fact, most of search user interfaces treated the search task as independent from the rest of the other ongoing user tasks. As far as we know, the *inside-in* approach proposed in this paper is an original contribution that can improve users performance.

VII. Concluding Remarks

This paper proposed a new perspective for looking at the way search user interfaces can be conceived for helping users to perform ancillary-search tasks on the Web. The work typically followed a user-centered design approach. Our initial motivation was to understand what makes searching on the Web so difficult to users. We found out that the predominant approach for searching based on *inside-out* approach is fine when users want to freely explore the information space. However, such approach presents several limitations when the users need to connect the results of search engines with tasks they are performing in another Web page. The proposed *inside-in* approach aims at reducing both execution and evaluation gulfs in the user interaction with search engines. Indeed, one of the key aspects of this approach is to provide a better integration of search results into existing Web pages, where users require complementary information to make their decisions.

Overall the *inside-in* approach is generic and can be implemented using current search engines such as Google or Yahoo! Nonetheless, it can also be implemented using search engines that are suitable to provide more focused and accurate results about data in a specific application domain. Our framework follows this latter approach as illustrated with the implementation of queries for searching co-authors in the DBLP. While looking up for co-authors might be perceived as a very narrow and specific search, it is noteworthy that it is relevant and frequent in the domains of scientific research, and also is a concern to a large population of researchers, students, teachers, and experts from research funding agencies. Moreover, such specialized characteristic can be tuned and adapted according to specific users' needs. Indeed, the main challenge here remains the identification of relevant queries that are suitable to help users to accomplish their tasks.

The tool implementing our approach provided with the framework allows users to: (a) launch a query by selecting a keyword directly in the web page, and (b) display the search results inside the current Web page as a kind of contextual help. In order to support the display of results, we have embedded into the framework an information visualization technique called IRIS. One of the interesting aspects of IRIS is that it is interactive, so users can explore the results and perform nested queries that are meant as ancillary-search tasks.

It is worthy to notice that IRIS is complementary to the *inside-in* approach proposed in the paper but it can also be used as an *inside-out* standalone tool. This aspect about the uses of IRIS became evident during the investigation of design alternatives for displaying the search results of co-authors. Indeed, the alternative shown in Fig 5.a displays the results in another tab of the browser, while the alternatives shown in Fig. 5.b and 5.c adapt the existing Web page to accommodate the results of the ancillary search.

The results obtained by a survey with 61 remote participants confirmed our first hypothesis: most users prefer to launch queries directly from the web page by selecting a keyword. This is not a new finding [16] but indicates that we are in the right path. As for the other three hypotheses, they were confirmed: users also prefer search results being displayed through an interactive visualization technique, located near the search keyword. Regarding location, users expressed to prefer the display of results in a way that does not change their context, this being achieved by two alternatives – displaying the results embedded in the web page, by augmenting it, or displaying them in a floating layer over the same web page.

With these results we also confirm that the semantic and articulatory distances of inputs (*execution gulf*) in the search task are reduced because searching is launched by clicking on a keyword displayed in the Web page. The semantic and articulatory distances of output (*evaluation gulf*) are also reduced in two of our designs (identified as Red and Green) because search results in both are placed in the same page.

Despite the promising results, we know that these are preliminary and there is much work to be done. We would like to measure the distances in the gulfs by performing experiments with direct observation methods. We also intend to proceed with the development of different input and output techniques for performing search tasks since our framework was developed aiming at such studies. The participants of our survey provided a rich set of comments that will allow us to plan further improvements in IRIS as well as to develop techniques targeted for different search tasks.

Future work should include empirical testing with users in a usability laboratory. This step would allow us to assess user performance when performing the tasks and collect more qualitative data via thinking aloud that would better explain the user experience factors that influence the use of information visualization techniques for displaying search results.

Acknowldegments

We are deeply grateful to the remote users that participated in the experiment. We also acknowledge the financial aid from CNPq, CAPES/COFECUB (project VIDAS 735-12) and FAPERGS.

References

[1] Bederson, B. B. Fisheye Menus. Proceedings of ACM Conference on User Interface Software and Technology (UIST 2000), pp. 217-226, ACM Press.

[2] Benford, S., Snowdon, D., Greenhalgh, C., Ingram, R, Knox, I., Brown, C. VR-VIBE: A Virtual Environment for Co-operative Information Retrieval. Computer Graphics Forum, 14(3):349-360, August 1995.

[3] Bouvin, N. O. Unifying Strategies for Web Augmentation. In: Proc. of the 10th ACM Conference on Hypertext and Hypermedia, 1999.

[4] Capra, R., Arguello, J., Scholer, F. 2013. Augmenting web search surrogates with images. In Proceedings of the 22nd ACM international conference on Conference on information & knowledge management (CIKM '13). ACM, New York, NY, USA, 399-408.

[5] Card, Stuart K.; Mackinlay, Jock D.; Shneiderman, Ben. Focus + Context. In: Card, Stuart K.; Mackinlay, Jock D.; Shneiderman, Ben. Readings in Information Visualization: Using Vision to Think. San Francisco, California: Morgan Kaufmann Publishers, 1999. p.307-309.

[6] Chau, M. Visualizing web search results using glyphs: Design and evaluation of a flower metaphor. ACM Trans. Manage. Inf. Syst. 2, 1, Article 2 (March 2011), 27 pages.

[7] Cugini, J., Laskowski, S., Sebrechts, M. Design of 3D visualization of search results: evolution and evaluation. In Visual Data Exploration and Analysis, pages 198-210, IST/SPIE, 2000.

[8] Dorneles, C. F., Gonçalves, R., dos Santos Mello, R. Approximate data instance matching: a survey. Knowledge Information Systems, 27(1): 1-21 (2011).

[9] Draper, G. M., Livnat, Y. and Riesenfeld, R.F. A Survey of Radial Methods for Information Visualization. IEEE Transactions on Visualization and Computer Graphics, 15(5):759-776, 2009.

[10] Firmenich, S., Winckler, M., Rossi, G. A Framework for Concern-Sensitive, Client-Side Adaptation. In Proc. of International Conference on Web Engineering (ICWE 2011), Paphos, Cyprus, June 20-24, 2011. Springer, LNCS 6757, pages 198-213.

[11] Hassan, A., White, R. W., Dumais, S.T., Wang, Y-M. Struggling or exploring?: disambiguating long search sessions. WSDM 2014: 53-62.

[12] Hearst, M. User Interfaces for Search, Chapter 2 of Modern Information Retrieval: The Concepts and Technology behind Search (2nd Edition), Addison Wesley, 2011.

[13] Heer, J., Robertson, G. Animated Transitions in Statistical Data Graphics. IEEE Transactions on Visualization and Computer Graphics, 13(6):1240-1247, 2007.

[14] Mann, T.M. Visualization of WWW-search results. In DEXA Workshop, pages 264-268, 1999.

[15] Marchionini, G. Interfaces for end-user information seeking. Journal of the American Society for Information Science, 43(2):156-163, January 1999.

[16] Marchionini, G. Exploratory Search: From Finding To Understanding. Comm. of the ACM, 49(4):41–49, 2006.

[17] McCrickard, S. and Kehoe, C. Visualizing search results using SQUID. Poster at 6th World Wide Web Conference, April 1997. Available at: http://people.cs.vt.edu/~mccricks/papers/mk97.pdf

[18] Norman, D. The Psychology Of Everyday Things. (1988). Basic Books; 1 edition (June 13, 1988). Basic Books. ISBN 978-0-465-06710-7

[19] Norman, D. A., Draper, S. W. (eds.) (1986): User Centered System Design: New Perspectives on Human-Computer Interaction. Hillsdale, NJ, Lawrence Erlbaum Associates.

[20] Roberts, J.C., Boukhelifa, N., Rodgers, P. Multiform glyph based search results visualization. In Proceedings of Information Visualization 2002, pages 549-554, IEEE, July 2002.

[21] Schwarz, J., Morris, M. R. Augmenting web pages and search results to support credibility assessment. CHI 2011: 1245-1254.

[22] Sebrechts, M.M., Vasilakis, J., Miller, M.S., Cugini, J.V., & Laskowski, S. (1999). Visualization of Search Results: A Comparative Evaluation of Text, 2D and 3D Interfaces. In Hearst, M.A. et al. (Eds.), Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp 3-10.

[23] Sutcliffe, A.G., Ennis, M. Towards a cognitive theory of information retrieval. Interacting with Computers, 10:321–351, 1998.

[24] Suzuki, E., Ando, S., Hirose, M., Jumi, M. Intuitive Display for Search Engines Toward Fast Detection of Peculiar WWW Pages. In. WImBI 2006. N. Zhong et al. (Eds.): WImBI 2006, LNAI 4845, pp. 341–352.

[25] Tversky, B., Morrison, J., Betrancourt, M. Animation: Can It Facilitate? Int. J. Human-Computer Studies, 57:247-262, 2002.

[26] Veerasamy, A. Heikes, R. Effectiveness of a graphical display of retrieval results. In Research and Development in information retrieval, pages 236-245, New York, 1997.

[27] Wilson, Max L. Evaluating the Cognitive Impact of Search User Interface Design Decisions. EuroHCIR 2011: 27-30.

[28] Winckler, M., Gaits, V., Vo, D-B., Firmenich, S., Rossi, G. An Approach and Tool Support for Assisting Users to Fill-in Web Forms with Personal Information. In Proc. of the ACM SIGDOC 2011, Pisa, Italy, October 3-5, 2011, ACM, New York, NY, USA, pp. 195-202.

Gathering Alumni Information from a Web Social Network

Gabriel Resende Gonçalves; Anderson A. Ferreira; Guilherme Tavares de Assis
Departamento de Computação
Universidade Federal de Ouro Preto
Ouro Preto, MG, Brazil
gbrll.rg@gmail.com, {ferreira, gtassis}@iceb.ufop.br

Andrea Iabrudi Tavares
Jasper Design Automation
Mountain View, CA, USA
andrea.iabrudi@gmail.com

***Abstract*—An undergraduate program must prepare its students for the major needs of the labor market. One of the main ways to identify what are the demands to be met is creating a manner to manage information of its alumni. This consists of gathering data from program's alumni and finding out what are their main areas of employment on the labor market or which are their main fields of research in the academy. Usually, this data is obtained through available forms on the Web or forwarded by mail or email; however, these methods, in addition to being laborious, do not present good feedback from the alumni. Thus, this work proposes a novel method to help teaching staffs of undergraduate programs to gather information on the desired population of alumni, semi-automatically, on the Web. Overall, by using a few alumni pages as an initial set of sample pages, the proposed method was capable of gathering information concerning a number of alumni twice as bigger than adopted conventional methods.**

***Keywords*-alumni information management; web social network; search engine; focused crawling; linkedin.**

I. Introduction

Nowadays, in academic environments, one of the major concerns of teaching staffs of undergraduate programs is analyzing how their students adapt to the professional life after their graduation. This concern exists, essentially, as a means to know if the curricular grating and the program syllabus of a given undergraduate program, passed onto its students, are managing to meet the needs of the current labor market.

Within this context, the definition of an efficient method to obtain professional data from alumni of undergraduate programs becomes necessary. According to Lousada and Martins [1], the traditional methods employed to gather alumni information do not have good collaboration rate, which lead to interrupt of gathering such a information. The problem of managing alumni information is recurrent in several universities. Thus, this work aims to propose a tool capable of semi-automatically retrieving professional data of alumni from the Web to help teaching staffs of undergraduate programs. The tool has methods for gathering relevant data about alumni from a social networks using a given available search machine.

Social networks are oriented either towards entertainment of its users or spreading professional data about their users. For the first purpose, there exist, for instance, MySpace[1], Twitter[2] and Facebook[3]. For professional purposes there exist, for instance, LinkedIn[4] and Bayt[5] [2].

Our tool uses a search engine for getting from LinkedIn a initial set of candidate pages. Next, the tool selects from the set of candidate pages the ones similar to the alumni population from a given under graduate program. To calculate the similarity, we use some pages gathered from LinkedIn as examples. Instead of providing data about the undergraduate program and institution of a alumni list, our tool receives as input a few LinkedIn alumni pages from this list. Furthermore, in order to perform effectively, our tool also receives as input the list of alumni's name from an undergraduate program.

In sum, the main contribution of this work is the definition of a new method performed by our tool that is capable of gathering information regarding alumni of a given undergraduate program. Moreover, this work presents an experimental comparative evaluation between our method and a traditional classifier in the retrieval of alumni pages from the web.

The rest of this paper is organized as follows. Section II presents works directly related to the goal of this work. In Section III, we describe our proposed method for gathering semi-automatically information on alumni of a given undergraduate program. In Section IV, we evaluate our method. Finally, in Section V, we present conclusion and perspectives for future work.

II. Related Work

Our proposed method, in this work, aims to perform a focused crawling of information on alumni on the web and extracts such a information. Based on that, the works related to this paper are divided in the following subsections: alumni information management and web pages focused crawling.

[1]http://www.myspacecom
[2]http://www.twitter.com
[3]http://www.facebook.com
[4]http://www.linkedin.com
[5]http://www.bayt.com

978-1-4799-6953-1/14 $31.00 © 2014 IEEE
DOI 10.1109/LAWeb.2014.17

A. Alumni Information Management

Taking into account that promoting management of alumni information is an important task to undergraduate programs of higher education Brazilian institutions, some studies were conducted to evaluate the benefits that such management may bring to the programs. Thus, several alumni monitoring programs have already been created in Brazil.

In [3], the authors conducted studies on the difficulties encountered in the management of alumni from educational institutions in Brazil. The authors introduce the concept of alumni within the Brazilian sphere. On that basis, alumni were categorized in graduates, students transferred to other schools, students that were dismissed from the institution and students that quit their programs. It was concluded that the monitoring concerning alumni is a means of assessing education outcomes of an institution so that improvements in its teaching methodology can be made. Furthermore, the work also highlights that, in order for the alumni management to be accurate, those must be considered according to the four previously mentioned categories.

In [4], the authors perform studies on benefits and potentialities that might be explored by an institution, when promoting alumni monitoring management. It is noteworthy that the challenges concerning the alumni information management does not happen solely in Brazil, but in a number of countries. As a conclusion, the authors mention that the alumni management provides better effectiveness of the institutional actions promoted by higher education institutions.

In the Federal University of Santa Catarina, a system was created to perform alumni tracking. The method contacts the alumni using a online portal where alumni enter and answer some questions [5]. Up to the moment of publication of the results, only about 6.8% of all alumni from the institution had contributed to the portal. On the other hand, regarding the alumni monitoring program from the environmental engineering program of the University of São Paulo, the institution achieved 79.3% of responses to emails that were sent to the alumni up to the publishing moment [6]; the explanation to such high response rate obtained by the referred program lies on the fact that the undergraduate program is not old, showing a number of only 140 alumni.

In the University of São Paulo, a system called Egressos-USP renders the management of information on the institution's alumni [7]. The data gathering is done through an online survey divided in four pieces. In the first piece, a alumnus must answer about its professional situation. In the second piece, the alumnus answers questions concerning the university structure. In the third piece, there exist questions about personal opinion on the labor market. In the fourth piece, the alumnus answer questions about "life goals" proposed in [8]. This system has obtained the collaboration of about 5% of the university alumni population, graduated in the last 10 years.

In the Department of Computer Science at Federal University of Viçosa, a survey was made on the professional profile of alumni of their undergraduate program[9]. Upon the survey, the computer science program had 357 alumni. The survey obtained the collaboration of 94 alumni, which represents about 26.33% of the total population of these alumni. In this case, the alumni population is relatively small, which made it viable for the research to be conducted through standard means: data was collected through an available form at the department website.

Overall, the traditional methods for obtaining alumni information are only effective for those programs that present a small number of alumni. In case of a larger alumni population, traditional methods may show inaccurate results, since the taken sample might be too small. Furthermore, such methods take weeks or even months to achieve the desired collection. In this work, alumni information is found through means of data publicly available on social networks from the alumni themselves. Since the process is done in a semi-automatic manner, it does not depend on the collaboration of the alumni. Moreover, the proposed method ensures to gather alumni information from a given program within a reduced period of time, depending on the limitations presented in Section III.

B. Web Pages Focused Crawling

Focused crawler aims to crawl web pages that are considered relevant to a specific user interest. There exist several works concerning focused crawling, involving proposed heuristics to such end [10], [11] and classification schemes [12].

Particularly, in [13], the authors propose a heuristic for focused crawling based on genre and content of the information contained on the web pages. Several experiments were conducted in order to demonstrate the effectiveness and efficiency of the proposed heuristic; some experiments were based on the information crawling from a few disciplines of the Computer Science program. The results showed that the proposed heuristic can reach a F1 value [14] greater than 0.92. In [15], the authors improved the heuristic efficiency by considering the link context of web pages.

In [12], the authors show a comparison between focused crawlers based on the SVM, Neural Networks and Naive Bayes classifiers. Among all three crawlers, the one based on SVM was the most effective one. Although experiments show that Naive Bayes is the worst choice within the three classifiers, this is the one that presents the lower cost for generating the classification model.

Unlike the mentioned studies, the proposed method in the present work does not make use of classifiers in order to determine the relevance of a page and does not correspond to a heuristic that can be applied in any context, depending

on the user needs on information. The proposed method focuses on retrieving only information of alumni from a social network, thus becoming more efficient and effective within such context.

III. Proposed Method

As previously mentioned, in this section, we describe the proposed method for the construction of our tool aiming gathering from the Web, semi-automatically, information on alumni of a given undergraduate program. Figure 1 presents the functioning architecture of the proposed method.

Notice that the method encompasses three main modules and two repositories. The first module, called *Searcher*, aims at searching, from a social network on the web, candidate pages to belonging to alumni from an undergraduate program, through a given available search engine. This module receives, as input, a list of the desired alumni's names. The second module, called *Filter*, aims at filtering, among the candidate pages retrieved by the first module, the ones that are in fact of alumni from an undergraduate program. The third module, named *Extraction*, aims at extracting, from the pages filtered by the second module, the data of the alumni (the alumni data). These data may be academic, professional or personal, depending on what is available within its content. The first repository, called *Pages Repository*, stores the pages from the initial set of samples, being yet incremented as the *Filter* module determines relevant pages of alumni. The second repository, named *Final Database*, corresponds to a database where the data on each alumnus is stored.

This section is organized as it follows. In subsections III-A, III-B and III-C, the *Searcher*, *Filter* and *Extraction* modules, we describe each module in detail.

A. Searcher

The first task, which must be performed by the tool based on the proposed method, is searching for candidate pages for the *Filter* module, which means determining what pages might belong to the alumni set of an undergraduate program. Those pages are retrieved from a social network, through public pages available on the web. In order to do so, the module receives as input the list of alumni's names from an undergraduate program.

We use LinkedIn as the social network for searching professional data on alumni. LinkedIn recently became a powerful professional contact network within the labor market. Nowadays, the network holds over 277 million users around the world, being Brazil the third greater country in number of records on the site, with a little more than 16 million users[6]. That represents around 8% of the total country population and 20% of all Brazilians with Internet access [16]. On average, 35% of LinkedIn users access the website daily and other 32% access it at least once a week.

[6]http://press.linkedin.com/about (as of Apr. 2014)

Figure 2: Example of Academic Data from a LinkedIn Page.

The social network obtains, on average, 2 new members per second [17]; in other words, LinkedIn grows at a fast pace with, approximately, 172.700 new users a day.

Another important LinkedIn feature is the fact that the network offers, in a reduced format, its user pages publicly on the web. That causes different existing search engines index a large number of its pages. In addition, the indexed reduced pages contains several data such as name, professional address and academic degrees.

Figure 2 shows academic data of a LinkedIn user. Notice that, there exist some data that can be irrelevant to the *Filter* module, as for instance the graduation date. However, data concerning the undergraduate program, program degree and institution are relevant and used by the method's *Filter* module.

LinkedIn has an Application Programming Interface (API) in order to search data from its users. This API were not used in this module because of searching limitations. First of all, the API must undergo authentication with a registered user profile on the network; after authenticating, it only allows the user to automatically visit and retrieve data from other users that are separated from such a user by few degrees of separation. Moreover, another limitation of the API is the number of available attributes for consult: the social network is more careful in exposing its data through API that enable extraction; therefore, LinkedIn provide more attributes on network's public pages available on the Web.

Thus, our tool becomes more effective to extract the desired information on users of the social network from the public pages available on the web, instead of consulting the social network itself. There exist several methods used to extract information from pages of social networks [18]. However, in order to retrieve them, one can use the API of a given available search machine. In this work, the API from Google[7], called Custom Search Engine (CSE), was used. Other works have already adopted such a API to perform crawling on the web [19]. Google's search engine holds a massive repository of indexed social network

[7]http://www.google.com

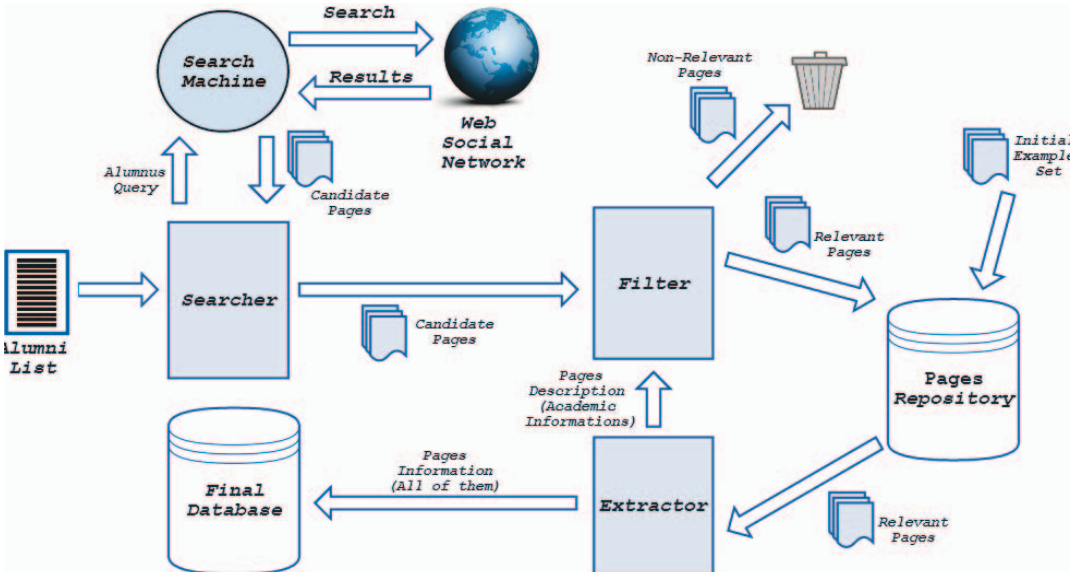


Figure 1: Functioning architecture of the proposed method

pages and that makes it a good tool when performing the assignment proposed on the *Searcher* module. On average, an undergraduate program features hundreds of alumni and, since the module searches for each alumnus, one must search via CSE the same number of alumni, for each undergraduate program. CSE API is limited regarding its usage, offering 100 daily free searches with 100 results per query.

Using the CSE, the *Searcher* module provides to the search engine a combination of the first, middle and last names of a given alumnus. This causes the proposed tool to be robust enough to retrieve user pages who do not use their full names on social networks. However, if we generate a large number of combinations, more results are retrieved by the queries, which might lead to large amount of candidate pages and a bottleneck in the runtime of the module. For instance, to search for the alumnus called ”Gabriel Resende Gonçalves”, the query includes the names ”Gabriel Resende”, ”Gabriel Gonçalves” and ”Gabriel Resende Gonçalves”.

B. Filter

The *Filter* module, the main module of the proposed method, aims at determining the significance of candidate pages, provided by the *Searcher* module, filtering the pages that in indeed belong to the alumni population of the desired program by the user. In order to do so, the module receives, as input, a initial set of sample pages as well as the candidate pages returned by the previous module. *Pages Repository* is started with such initial set of sample pages (small set of alumni pages initially obtained) so that the *Filter* module is able to start the classification, and carry on being incremented as the module recognizes a new page as relevant one, which means it is related to an alumnus from an undergraduate program.

In this module, we calculate the similarity among pages using only their academic data (i.e., data about undergraduate program, attended institution and degree). Thus, the method does not attempt to determine whether a certain candidate page indeed concerns the sought alumnus, but if the page belongs to the population described on the set of sample pages.

There are several classifiers based on supervised machine learning [20] that obtain fine results to page or text classification [21]. Most of them can be used in order to determine whether a candidate page belongs or not to the population set. One of the possible classifiers to be adopted is the *Naive Bayes*, which consists of a method that calculates the probabilities of page instances belonging to the predefine classes on the training sets; those probabilities are, usually, calculate through the Bag of Words model, based on word occurrence counting. Naive Bayes assumes that the terms of each classifying instance is independent of one another, and that it bad for databases which do not exhibit such a feature [22]. Another classifier which could also be used is the SVM that attempts to find the hyper plane that best separates the instances of the training set into smaller subsets. However, since the set of examples is incremental in this work, the cost for generating a new hyper plane, every time an instance is classified, might be high, which makes this an unusable method in terms of time. Nonetheless, another technique that may be adopted to classify candidate pages is the application of a heuristic function that relates a given candidate page to the training set. This function might be, for instance, the cosine similarity function [14], which

determines the similarity between two vectors based on their relative opening angle in the vector space; in this case, each vector would be defined on the page terms. For this technique, it is necessary to determine a similarity threshold for checking the similarity among the pages.

Beyond using classifiers and heuristic functions, we may evaluate a candidate page by means of its similarity regarding a set of predefined terms. In this case, the terms must be defined by a specialist and represent different information about a given undergraduate program such as the program and institution names. However, such specification may be a laborious task to a specialist, since these types of information may show many variations and, in order the tool presents a good result, most of them must be predicted. Another difficulty is the fact that the specified terms are not unitary, since an ambiguous term cannot have the same meaning as an unique term. For instance, in the case of the ”Computer Science” program, the unique terms ”Science” and ”Computer” do not have the same meaning of ”Computer Science”. Thus, the values that represent the significance of the terms must be empirically determined, and that also consists of a laborious task.

Hence, for our *Filter* module, we propose a new strategy to select the candidate pages returned by the *Searcher* module. The strategy does not depend on the user for determining the terms, instead, an initial set of positive page examples must be manually defined. The terms are extract from these examples. Moreover, in order to improve the *Filter* module, our proposed strategy requires the specification of a γ value that is the minimum percentage of pages in which a certain term must figure to take in account on the similarity calculation. For instance, if one defines the γ value as 0.15, the term must figure in 15% of the pages in order to be considered.

Usually, a classifier would apply a similarity function in order to relate all terms from a given candidate page with the terms from the page examples. In this work, the terms are separated in three groups: undergraduate program name, institution and program degree. The heuristic function is applied separately for each of the three types of terms and the final result is given by an average of the three obtained similarities.

The threshold used to determine whether a candidate page concerns an alumnus of a given program consists in the lower value obtained by applying the similarity function to determine the resemblance of each page of the set of sample pages with the other ones since, in the such set, all pages belong to the desired alumni.

The selected similarity function to build the relation between candidate pages given by the *Searcher* module and the set of page examples was *Cosine Similarity* [15]. Preliminary experiments were conducted using other functions, such as Jaccard Distance function; however, the results were not satisfactory.

In order to calculate the *Cosine Similarity*, it is necessary to create a n-dimensional vector for each term type (undergraduate program, institution and degree), where n is the number of terms with frequency over the pre established γ value. After defining the vector, calculation is made according to Equation 1.

$$Cosine = \frac{A.B}{||A||||B||} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}} \quad (1)$$

where A is the vector of terms from the page examples and B is the vector of terms from the evaluated candidate page.

C. Extraction

The *Extraction* module is responsible for extracting, from the HTML pages returned by the *Filter* module, relevant information of alumni from an undergraduate program. In this work, since the page structures from a same social network (LinkedIn) are single, i.e., all pages feature the same pattern, our strategy for extracting the data is based on regular expression [23] manually defined. Table I shows the information along with their respective regular expressions.

This module extracts academic, professional and/or personal information from the filtered pages by the *Filter* module. In this work, according to the architecture shown in Figure 1, this information have two purposes: the first is the storage on the *Final Database* repository, in order to further analyzing the obtained results (in this case, all information is stored); the second is defining and enhancing the description of the set of sample pages used by the *Filter* module (in this case, only academic information is required).

IV. Experiments

In this section, we describe the experiments and discuss the results, in order to validate the proposed method for constructing the tool that gathers on the Web, semi-automatically, professional data on the alumni of an undergraduate program. The section is organized as follows. In Subsection IV-A, we present experiment setup: alumni lists, baseline and evaluation metrics. In Subsection IV-B, we present and discuss the obtained results.

A. Experimental Setup

In this subsection, we describe the experimental setup and inputs of our method. The experiments for validating our proposed method of semi-automatic gathering professional data on alumni of an undergraduate program (see Figure 1) consisted on the conduction, for each alumni list, of 10 distinct runs and, for each run, a random set of 15 initial page examples are used as training examples.

Table I: Regular Expressions Used by *Extraction* Module

Information	Regular Expressions
Given Name	(.*?)
Family Name	(.*?)
Jobs	\n(.+)\n<span.*at.>.+\n(?:<a.+><.+summary">)?(.+)
Currently Job Location	<.*locality.*>\n*(.*?),.*\n
Programs	<span.*major.*>\n(.*)
Program's Degrees	<span.*degree.*>(.*)
Schools	<div.*education.*>\n<h3.*summary fn org.*>\n(.+)\n</h3>

Alumni Lists

As previously mentioned, our proposed method receive, as input, a list of alumni's names from an undergraduate program. We perform experiments with five alumni lists, available on the web, concerning the following undergraduate programs: Computer Science of the Federal University of Minas Gerais (UFMG) [8], Metallurgical Engineering of the Federal University of Ouro Preto (UFOP) [9], Chemistry of the University of So Paulo (USP) [10], Computer Science of the USP [11], and Computer Science of the Catholic Pontifical University of Paran (PUC-PR) [12]. Table II shows the number of alumni available in each list.

Table II: Population size of each alumni list.

Alumni list	Population size
UFMG	1,542
UFOP	1,579
USP - Comp. Sci.	1,259
USP - Chemistry	900
PUC-PR	812

Baseline

In this work, for evaluating our proposed method that semi-automatically gathers alumni information of an undergraduate program, we compare it with the *Naive Bayes* classifier. Naive Bayes generates a classification model with low cost, since it only counts the term occurrences according to the *Bag of Words* model. Naive Bayes considers the program, institution and degree terms mutually independent. Furthermore, others classifiers such as *SVM* and *Neural Network* need for training set with, at least, two training sets. Our proposed method uses only a set of pages with true examples.

Evaluation Metrics

Typically, works related to the information retrieval area are evaluated by precision, recall and F-mean metrics [14]. However, for this work, precision and the number of retrieved relevant pages were adopted. Precision can be calculated as demonstrated in Equation 2. The reason why the recall value is not used was the impossibility of obtaining the exact number of alumni pages available on a social network.

$$Precision = \frac{RelevantPages \cap RetrievedPages}{RetrievedPages} \quad (2)$$

B. Experimental Evaluation

γ Determination

As previously mentioned, in order to improve the process of filtering candidate pages performed by the *Filter* module, the strategy to such filtering requires the specification of a γ value, which corresponds to the minimum percentage of pages in which a given term must feature. Therefore, initially, several tests were conducted in order to determine the best γ value to be used in our experimental evaluation. The results are the average value obtained for each γ value on 10 runs.

Figure 3 shows, for different γ values, the precisions and the number of relevant pages obtained by our method varying the γ value from 0 to 1. Notice that a great value for γ was 0.2 since, for this value, a great number of relevant pages was obtained, maintaining precision at a satisfactory level. Thus, we use such γ value in our experimental evaluation, i.e., only the terms, that features at least 20% of the set of page examples, are considered.

Experimental Results

Table III shows, for each alumni list, the most frequent terms encountered on the performed experiments. This table shows the terms concern, for each alumni list, the test that obtained the greater precision between all 10 runs.

Table IV shows the results gathered when performing experiments considering, respectively, our proposed method and the baseline. This table contains the average precision and the average number of found relevant pages, for each alumni list, considering a 99% confidence interval.

Notice that the baseline has retrieved a greater number of pages; however, the precision obtained is much inferior when in comparison with our proposed method. Considering all alumni lists, our proposed method obtained the average precision from 0.83 to 0.91, while the best result for the baseline was 0.65 on average precision. Unlike our proposed

[8]http://dcc.ufmg.br/dcc/index.php?option=com_content&view=article&id=274&Itemid=8; (as of Apr. 2014)

[9]http://www.em.ufop.br/exalunos; (as of Apr. 2014)

[10]http://www.iqsc.usp.br/acad1/egressos/app/graduacao/lista/index; (as of Apr. 2014)

[11]http://www.ime.usp.br/c̃gmac/ex-alunos/res.html; (as of Apr. 2014)

[12]http://www.pucpr.br/graduacao/cienciacomputacao/egressos_curso.php; (as of Apr. 2014)

Table III: Most frequently terms for γ = 0.2 using cosine similarity function.

Alumni list	Program	School	Degree
UFMG	”ciencia da computacao” ”computer”	”ufmg”, ”universidade federal de minas gerais”	”ma”,”bachelor” ”bacharel”, ”bs”
UFOP	”engenharia”,”metalurgia”	”universidade federal” de ouro preto”	”engenheiro”
USP - Comp. Sci.	”ciencia da computacao” ”computer”	”usp” ”universidade”	”bachelor” ”master”
USP - Chemistry	”quimica”	”usp”, ”universidade	”bacharel”, ”doutor”
PUC-PR	”bacharel”	”ciencia da computacao”	”pontificia universidade” catolica do parana”, ”puc”

Table IV: Number of Pages Retrieved and Precision Results For Proposed Method and Baseline.

	Pages Retrieved		Precision	
Alumni list	Proposed Method	Baseline	Proposed Method	Baseline
UFMG	127 $\pm$ 6.04	381 $\pm$ 85.01	0.919 $\pm$ 0.01	0.368 $\pm$ 0.08
UFOP	85 $\pm$ 3.86	242 $\pm$ 64.21	0.839 $\pm$ 0.01	0.554 $\pm$ 0.06
USP - Comp. Sci.	155 $\pm$ 23.13	237 $\pm$ 89.97	0.86 $\pm$ 0.02	0.658 $\pm$ 0.14
USP - Chemistry	70 $\pm$ 6.297	129 $\pm$ 78.087	0.85 $\pm$ 0.02	0.593 $\pm$ 0.21
PUC-PR	34 $\pm$ 6.29	70 $\pm$ 57.28	0.893 $\pm$ 0.09	0.508 $\pm$ 0.13

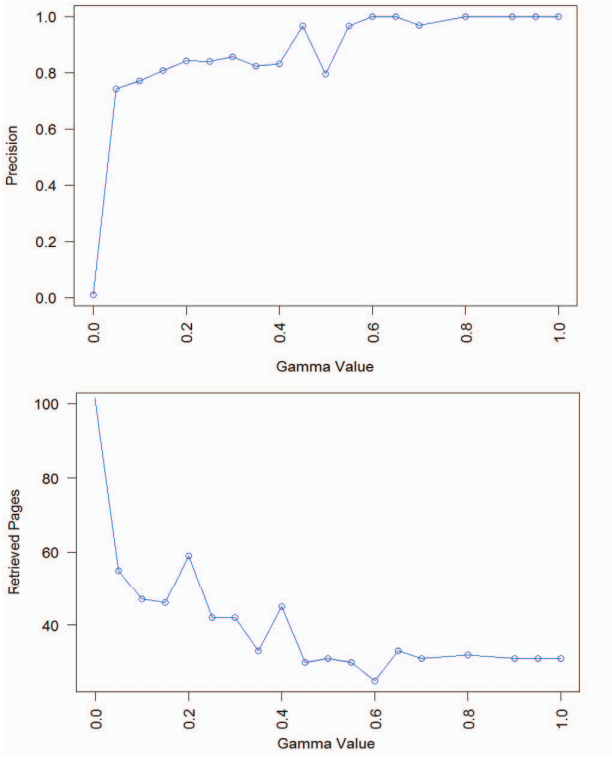


Figure 3: Results of γ Variation on Precision (Top) and Number of Retrieved Pages (Bottom).

method, the baseline results were too sensitive towards the initial set of training pages; therefore, the experiments showed significant variation on the results. That can be observed through means of the confidence interval on Table IV.

Estimating the coverage of LinkedIn

As we may not determine the recall value on a domain where the exact number of alumni pages is unknown, we perform another experiment for estimating the percentage of true alumni pages on LinkedIn. We manually search for alumni pages that are not found by our method in our experimental evaluation. For each alumni list, 50 names was randomly chosen. The Table V shows the percentage of pages that are found in this search. We can notice that, except for the experiment of Metallurgical Engineering from UFOP, a few alumni, who have pages on LinkedIn, were not found by our method. For Metallurgical Engineering from UFOP, the diversity in the course name filled in the pages leads to a poor performance.

Table V: Percentage of alumni pages found in the manual Search.

Alumni list	Percentage of pages
UFMG	0
UFOP	0.22
USP - Comp. Sci.	0.14
USP - Chemistry	0.10
PUC-PR	0.12

Descriptive analysis of the results

After gathering information on alumni of the 5 undergraduate programs considered in the performed experiments, we may inserted the data into the final repository (*Final Database*) and analyze their information. The repository was populated based on the test that showed greater precision for each one of the undergraduate programs.

Figure 4 shows the alumni distribution per year of graduation regarding the alumni of the 5 undergraduate programs adopted. Distributions regarding alumni from all programs were similar. One can observe the high concentration of

alumni in the years from 2000 to 2010. That can be explained by the fact that such alumni, newly graduated, are looking for new jobs. Hence, many seek to promote their professional resume on the Web and LinkedIn is a great social network for such purpose. Another relevant factor is the fact that LinkedIn network was created in the year of 2002; moreover, people who joined the labor market before 2003 could, at that time, not have interest and/or ease to engage in social networks.

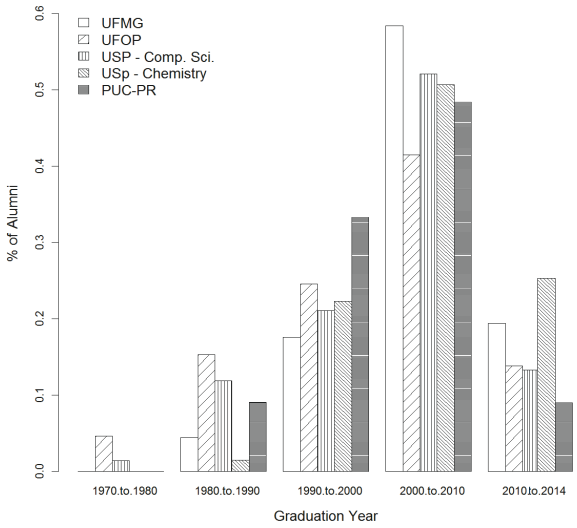


Figure 4: Graduation year graphic of all experiments.

For exemplification purposes, Figures 5 and 6 show location of the current employment of the alumni of all 5 undergraduate programs considered in the performed experiments. Images were generated with the aid of the maps API from Google Maps[13]. We can notice, based on Figures 5 and 6, a high concentration of alumni in the city of origin of their program and, with the exception of alumni in Computer Science from USP, all other ones are located, entirely, in the American continent.

V. Conclusion

Our proposed method for semi-automatic gathering professional data on alumni of an undergraduate program makes use of a list of alumni names and an initial set of relevant pages. At first, the method automatically performs searches for pages within a social network using a search machine, which are candidates to belonging to the alumni. Subsequently, the method uses a proposed strategy, based on similarity metrics, in order to determine the relevance of the candidate pages and, thus, retrieve only the ones that concern the alumni of the desired program.

[13]http://maps.google.com

Once prospective tool users, made from the proposed method, belong to teaching staffs of undergraduate programs, acquiring the necessary list of alumni names is not a problem. Obtaining the initial set of sample pages might come to be the greater complicating factor in the method's functioning since, in any social network, manually acquiring some pages may be a laborious task, even in small numbers.

In our experimental evaluation, considering alumni lists from five different undergraduate programs, our proposed method was capable of gathering information on them, within the social network LinkedIn, with satisfactory precision. The experiments show that the proposed method was able to find a great number of alumni pages. For undergraduate programs with over 1.000 alumni, the method was able to find, on average, 7.5% of alumni. Particularly, for alumni in Computer Science from USP, the method was able to find 12.2%. By using conventional methods for gathering information about alumni of undergraduate programs with a high number of alumni, the feedback rate is around 6% of the alumni population [5]. Moreover, conventional methods usually take days or even months to obtain such results, while our proposed method performs in few minutes, depending on limitations of the search engine API.

As future work, we intend to improve our method for gathering alumni pages from the web without providing an initial set of page examples. Another important work is the proposal of a strategy that enables the non necessity of a list of alumni names, in order to not restrict the use of the tool to only the users who have access to the this list of a certain undergraduate program. Furthermore, we intend to experiment the method on other purposes, such as retrieving relevant pages on a specific topic or individuals from a given group that is not from the academic sphere.

Acknowledgment

This research is partially funded by Fapemig (Foundation for Research Support of the State of Minas Gerais). Furthermore, this research was carried out on the GAID/UFOP Laboratory.

References

[1] A. C. Z. Lousada and G. d. A. Martins, "Alumni as a source of information to management accounting courses (in portuguese)," *Journal of Accounting and Finance*, vol. 16, no. 37, pp. 73–84, 2005.

[2] N. B. Ellison *et al.*, "Social network sites: Definition, history, and scholarship," *Journal of Computer-Mediated Communication*, vol. 13, no. 1, pp. 210–230, 2007.

[3] M. D. Pena, "Alumni monitoring: Conceptual analysis and its application in brazilian educational context (in portuguese)," *Technological Education of Belo Horizonte*, vol. 5, no. 2, pp. 25–30, 2000.

[4] L. S. Michelan, C. A. Harger, G. Ehrhardt, and R. P. O. Moé, "Alumni management in higher education institutions: Possibilities and potential (in portuguese)," *IX International Colloquium on University Management in South America*, 2011.

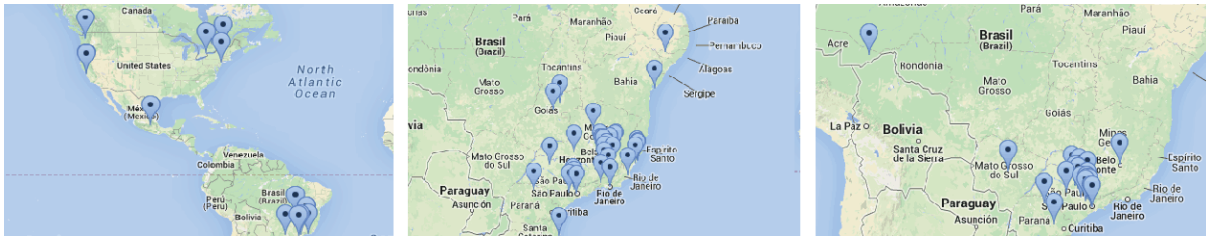


Figure 5: Alumni localization. UFMG on left, UFOP-Met. Eng. on center and USP-Chemistry on right.

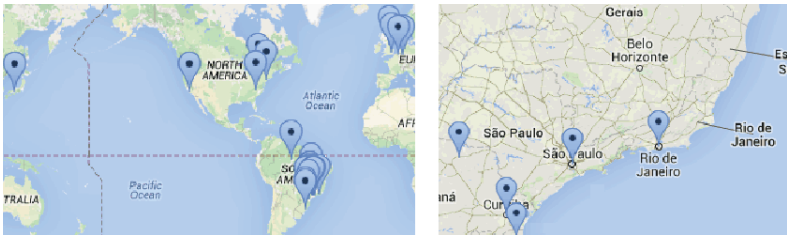


Figure 6: Alumni localization. Usp-Comp. Sci. on left and PUC-PR on right.

[5] J. M. Silva, R. d. S. Nunes, and A. d. L. Jacobsen, "The alumni monitoring program of the Federal University of Santa Catarina: The profile of students definition in the period 1970-2011 (in portuguese)," *IX International Colloquium on University Management in South America*, 2011.

[6] R. P. Morgado, C. G. Geroto, and A. C. G. Ramalho, "Course evaluation and professional status of the environmental management program ESALQ/USP (in portuguese)," *Electronic Journal of Master in Environmental Education*, vol. 27, 2013.

[7] U. O. Media, "Research reveals alumni profile of USP courses (in portuguese)," Retrieved April 12, 2014, from http://www.usp.br/imprensa/?p=31718, 2013.

[8] E. L. Deci and R. M. Ryan, *Self-Determination*. Wiley Online Library, 2010.

[9] J. L. Imbrizi, F. G; Filfo, "Alumni research - conclusive review (in portuguese)," Retrieved April 12, 2014, from http://www.dpi.ufv.br/arquivos/diversos/pesqegressos.pdf, 2003.

[10] G. Pant, K. Tsioutsiouliklis, J. Johnson, and C. L. Giles, "Panorama: Extending Digital Libraries with Topical Crawlers," in *Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital Libraries*, Tuscon, AZ, USA, 2004, pp. 142–150.

[11] P. Srinivasan, F. Menczer, and G. Pant, "A General Evaluation Framework for Topical Crawlers," *Information Retrieval*, vol. 8, no. 3, pp. 417–447, 2005.

[12] G. Pant and P. Srinivasan, "Learning to Crawl: Comparing Classification Schemes," *ACM Transactions on Information Systems*, vol. 23, no. 4, pp. 430–462, 2005.

[13] G. T. De Assis, A. H. Laender, M. A. Gonçalves, and A. S. Da Silva, "A genre-aware approach to focused crawling," *World Wide Web*, vol. 12, no. 3, pp. 285–319, 2009.

[14] R. Baeza-Yates, B. Ribeiro-Neto *et al.*, *Modern information retrieval*. ACM press New York, 1999, vol. 463.

[15] V. Mangaravite, G. T. Assis, and A. A. Ferreira, "Improving the efficiency of a genre-aware approach to focused crawling based on link context," in *Proceedings of the Eighth Latin American Web Congress (LA-WEB)*. IEEE, 2012, pp. 17–23.

[16] IBGE-Brazil, "Internet access and possession of mobile cell phone(in portuguese)," Retrieved April 19, 2014, from http://loja.ibge.gov.br/pnad-2011-sintese-dos-indicadores.html.

[17] L. Rao, "Linkedin now adding two new members every second," Retrieved April 19, 2014, from http://techcrunch.com/2011/08/04/linkedin-now-adding-two-new-members-every-second.

[18] E. Ferrara, P. De Meo, G. Fiumara, and R. Baumgartner, "Web data extraction, applications and techniques: a survey," *arXiv preprint arXiv:1207.0246*, 2013.

[19] M. Allauddin and F. Azam, "Service crawling using google custom search api," *International Journal of Computer Applications*, vol. 34, no. 7, 2011.

[20] S. B. Kotsiantis, "Supervised machine learning: a review of classification techniques." *Informatica (03505596)*, vol. 31, no. 3, 2007.

[21] D. D. Lewis, "Naive (bayes) at forty: The independence assumption in information retrieval," in *Machine learning: ECML-98*. Springer, 1998, pp. 4–15.

[22] I. Rish, "An empirical study of the naive bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, no. 22, 2001, pp. 41–46.

[23] V. Alfred, "Algorithms for finding patterns in strings," *Handbook of Theoretical Computer Science: Algorithms and complexity*, vol. 1, p. 255, 1990.

PanView: An Extensible Panoramic Video Viewer for the Web

Cassio E. dos Santos Jr, Jessica I. C. Souza, Virginia F. Mota,
Guilherme S. Nascimento, Guilherme S. Gorgulho, Arnaldo de A. Araujo
Department of Computer Science
Universidade Federal de Minas Gerais
Belo Horizonte, Brazil
{cass, jessicaione, virginiaferm, gsnasc, gsad, arnaldo}@dcc.ufmg.br

***Abstract*—In this paper we present a panoramic video viewer for the web. We propose PanView, an open-source video-based panoramic viewer that provides a greater immersion of filmed environments. The main advantage of consider panoramic video is that users can pan or tilt virtual cameras in the recorded scene, allowing him/her to focus on information presented in different angles or moments, thus, allowing him/her to access more information than in ordinary videos or in static images. PanView is fully extensible with easy customization using user coded modules and is implemented using modern web-browser standards, which reduces the computational requirements. To motivate the use of PanView, we present a performance comparison considering a panoramic viewer based on Adobe Flash Player. The applicability of PanView is shown in two projects: virtual tour on historical cities and analysis of a railroad network.**

I. Introduction

Panoramas consist in representing the surroundings of a scene in a single image, achieving, this way, several advantages over ordinary single view methods to acquire images. Panoramas allow users to generate images from specific views of the scene at the same time that other users are able to choose other views. The development of advanced devices and algorithms to build panoramas leverage the growth of services based on panoramic videos and images. The Google Street View [1], for instance, allow users to see panoramic images from different places around the world.

Among the different kinds of panoramas, panoramic videos are preferable since most applications, such as street view tours, remote meetings, or e-meeting, surveillance; require temporal information along with the image of the scene. The main advantage of considering panoramic video rather than regular videos is that users can pan or tilt virtual cameras in the recorded scene, allowing them to focus on a specific event of the video that would rather be lost in a regular camera with fixed view. In an e-meeting scenario, for instance, an attendee can focus on the presentation screen while other attendees can focus on the speaker or on the audience.

There are several challenges that hamper the use of panoramic videos, such as the large amount of video footage often required to analyze, store or index; and the lack of a common platform to visualize, analyze or distribute panoramic videos. Considering the aforementioned challenges, we propose *PanView*, a extensible video-based panoramic viewer.

Fig. 1: PanView interface. The map in the top-right corner and the control buttons on the bottom are drawn using user modules, which demonstrate the extension capabilities of PanView.

Figure 1 illustrate the extensibility of PanView regarding the add-in of a map to display geo-location data and control buttons. PanView is an open-source[1] and extensible framework, designed to be customized for a large set of applications.

The framework is organized following a microkernel architecture, composed by a *core* and several *user* modules. The *core* module includes common functionalities that are useful for different applications, such as draw the panoramic video and handle basic mouse inputs. *User* modules enable, for instance, components to be drawn in the panoramic video interface, such as a map to display geo-location data or on-screen control buttons. Since user modules are independent of each other, including or removing the file of a user module in a web-page code is enough to enable or disable that user module. Moreover, PanView is coded using modern web standards, such as WebGL and HTML5, ensuring, this way, higher performance when compared to panoramic viewers based on Adobe Flash Player[2].

The remainder of this paper is organized as follows. We discuss related works in Section II. Section III presents PanView. As motivation, we compare PanView with a public and

[1]PanView will be available at http://www.npdi.dcc.ufmg.br/PanView
[2]http://www.adobe.com/products/flashplayer.html

978-1-4799-6953-1/14 $31.00 © 2014 IEEE
DOI 10.1109/LAWeb.2014.19

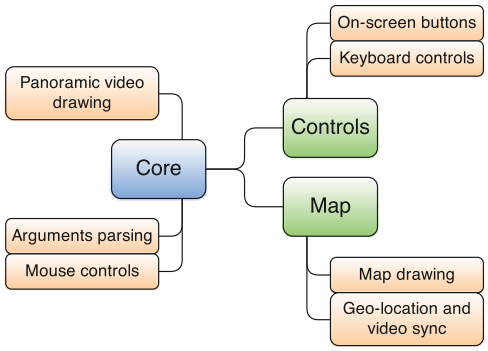


Fig. 2: PanView components. The *core* is the main module and it is responsible for draw the panoramic video, parse the input arguments and handle mouse inputs. The *controls* module handles keyboard inputs and on-screen buttons. The *map* module draws the map and synchronizes the geo-location with the panoramic video.

free panoramic viewer based on Adobe Flash Player. Examples of applications that use PanView are depicted in Section V. Finally, we conclude with final remarks and future directions in Section VI.

II. Related Works

There are several works in the literature describing approaches to assemble images acquired from different angles into a single continuous panoramic image. Examples of these works include match correspondent pixels in overlapping parts of the images to estimate warping parameters [2], [3], and join accelerometer and gyroscope data in mobile devices while capturing the surroundings of the scene [4]. Regarding panoramic videos, the common approach to assemble images consists in fasten cameras in a rigid structure to estimate warping parameters offline [5], [6]. Then the same transformation is applied to every frame in the captured video.

Panoramic videos draw attention in applications such as e-meetings [5], [6], [7], street view [1], [8], [9] and surveillance [10], [11]. Google Street View [1] is the most popular panoramic viewer available, but it lacks support to videos and present few customization options. Salado Player[3] provides the source code, thus, it allows more customization options, but still lacks video support. A video-based panoramic viewer built using Adobe Flash Player is krpano[4], however its source code is not public available. Kolor Eyes[5] is an advanced player for panoramic videos with versions for smartphones and desktops. It is a free application, although the source code is not available. The web version of Kolor Eyes is built on HTML5 and WebGL, but its use is limited to their own hosting service. Other websites can embed the hosted videos, which will be executed in Kolor Eyes web player with few customization options.

[3]http://openpano.org/
[4]http://krpano.com/
[5]http://www.kolor.com/360-video/kolor-eyes-player

Code 1: Required code to insert the PanView in a web-page.

```
<script src="panview_core.js"/>
<script src="plugins/panview_controls.js"/>
<script>
var options = { /* some player options */
        video: "video.mp4",
        mouseControls: true,
        keyControls: true,
        buttonControls: true,
        autoplay: true,
        autoresize: true,
        fov: 80,
        pole: {x: 0, y: 0, z: 0}
};
var pano = new PanView(div, options);
</script>
```

More examples of panoramic viewers can be found at PanoTools website[6], although most of them focus on high resolution panoramic images and do not support videos. For instance, Pano2VR, Pannellum and Leanorama, which are listed in the PanoTools website, are implemented in HTML5. None of them have support for videos. Moreover Pano2VR is not open-source while Leanorama has been discontinued.

Our work differs from the aforementioned works considering that our proposed panoramic viewer support panoramic videos, is open-source, presents easy customization using user modules and is implemented using modern web-browser standards, such as HTML5 and WebGL, which reduce the computational requirements.

III. PanView Architecture

The architecture of PanView follows a microkernel design pattern [12] and consists in a single *core* and several *modules* that can be written by users to extend basic functionalities. The core provides common functionalities to applications that require a panoramic viewer, such as parse user arguments, draw the panoramic video in the web-page and control basic input. The modules implement other functionalities such as map preview, button control and keyboard shortcuts. Each user module is independent from the others and it is added to the core when its file is included in the web-page source code. Then, a small code, illustrated in Code 1, is required to configure and draw PanView in the web-page. An overview of PanView architecture is presented in Figure 2.

In the following sections we describe the core, in Section III-A, and two user modules: *controls*, described in Section III-B, and *map*, described in Section III-C.

A. Core

The core consists in a single JavaScript file that holds the main class definition. The following functionalities are included in the core: draw each frame of the panoramic video in the web-page, control which part of the video sphere is

[6]http://wiki.panotools.org/Panorama_Viewers

Fig. 3: Sample frame from a panoramic video in spherical projection. Horizontal coordinates of pixels are mapped in longitude in the video sphere while vertical coordinates of pixels are mapped in latitude.

being draw, and handle basic mouse input. The mouse input control is included in the core since most applications require mouse navigation and the variables that control the view angles of the video sphere are closely related to the mouse control. To facilitate the correction of the navigation axis in the mouse control, the north pole of the video sphere can be adjusted by indicating Euler rotation angles of the sphere. This way, videos acquired using cameras in different rotations can be corrected in the parameters of the PanView.

The panoramic video is drawn considering spherical projection, therefore, the input video must be presented in P pixels height, mapped to latitude coordinates of the video sphere, and $2 \times P$ pixels width, mapped to longitude coordinates. An illustration of a panoramic image is present in Figure 3. We choose spherical mapping to be implemented in the core since it is more common to find applications to generate panoramic videos in this type of projection than in other (cubic or hexagonal, for instance). However, other projections can be implemented in user modules.

B. Controls Module

The controls module allows the user to interact with the panoramic video using keyboard inputs and buttons on the screen. The implemented actions are play/pause and the ones already possible to be executed with the mouse input, which are zoom in/out and camera rotation. The on-screen buttons can be selected using the tab key and activated using the *enter* key. Functions are available to disable or enable the keyboard controls and hide or show the buttons. The controls work with the most common web-browsers[7].

PanView default on-screen buttons are highly customizable and present effects on mouse hover, tab focus and when they are activated. A css style sheet is included allowing easy customization. It is possible to completely personalize the on-screen buttons with new images and css attributes. Each action is implemented as a separated function thus the keyboard controls and the on-screen buttons share the same action

[7]Google Chrome (http://www.google.com/chrome) and Mozilla Firefox (http://www.mozilla.org/firefox)

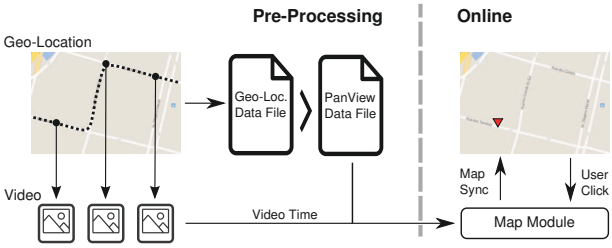


Fig. 4: Overview of the two processing steps of the map module. The panoramic video is composed by the video file and the geo-location file. On the pre-processing step, the geo-location file is converted to a structured file compatible with PanView. During the online step, the map module synchronizes the marker on the map according to the current video time. If the user performs a click on the map, the map module synchronizes the video and the marker according to the new position.

functions. It is possible to create new actions and buttons or change the default ones easily with simple modifications in the module source code.

C. Map Module

The map module draws a small map in the upper-right corner of the panoramic video (illustrated in Figure 1) which is used to display geo-location data associated to the panoramic video. A marker in the map indicates the geographic location of the frame being displayed in the panoramic video in the moment when that frame was captured, thus, it is required to update the marker for every frame of the video.

To avoid successive updates of the marker in the map, specially when the frame rate of the video is high, we employ a grid-based approach as follow. The geographical coordinates associated to each frame of the video are divided in cells of a grid and the marker is updated only once for each cell.

There are two processing steps in the map module: a *pre-processing* step, which converts the geo-location data to a structured file containing the grid information; and an *online* step, which draws the map and updates the marker and the time of the video when the user clicks in some location in the map. Figure 4 presents an overview of those two processing steps.

1) Pre-Processing: in this step we convert the geo-location data file (such as KML or GeoRSS) in a file that is compatible with PanView. The conversion script of the geographic meta-data file has certain procedures, adopted to associate the time in the panoramic video with the position of the marker in the map. The meta-data has a list of geographic coordinates and, associated with it, the time of each transition from a geographical location to another. The script read the input geographic data and converts to a structured data file which can be read by PanView.

2) Online: the online step consists in the user module, which receives the pre-processed structured file containing

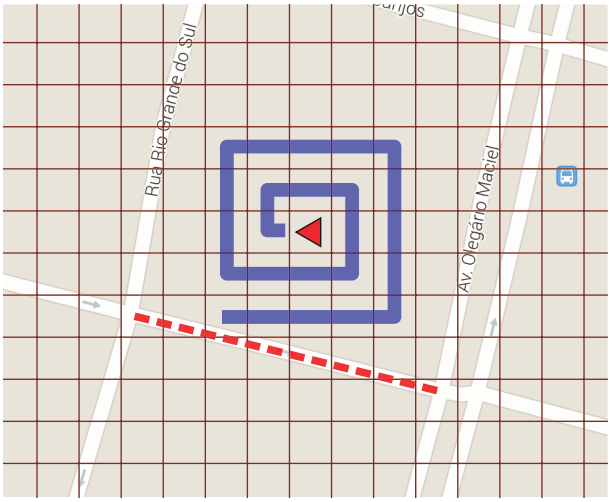


Fig. 5: Process to correct clicks of the user (triangle) out of the path described by the coordinates associated to the panoramic video (dotted line over the street). The cells around the clicked area are visited in a spiral pattern until it reaches a cell which contains coordinates of the panoramic video or the maximum number of cells are visited.

the grid data and draws the marker in the map. Then, the marker is updated in the map each time that the geo-position of the current frame being displayed in the panoramic video corresponds to a new cell in the grid. To change the time of the video whenever the user click in a geo-location in the map, we search in the neighborhood of the clicked cell for one that contains a least one time interval. The search is performed following a spiral pattern as illustrated in Figure 5. This way, the number of cells visited and the cell size determine the maximum error allowed between the click of the user and the coordinates of the video.

IV. Performance Evaluation

In this section we compare PanView with krpano, an Adobe Flash Player panoramic viewer available in the web. Our goal in this experiment is to motivate the use panoramic viewers built using HTML5/WebGL instead of Adobe Flash Player, which correspond to the majority of available tools[8]. Although krpano provides a HTML5/WebGL version, at the time of submission of this work the available version did not support panoramic videos.

To compare krpano with PanView, we consider the mean and 95% confidence interval of the number of Frames Per Second (FPS) that each panoramic viewer presents when displaying panoramic videos in different resolutions. To calculate the FPS, we use Fraps tool[9]. The krpano is available in demo and premium versions. The only difference between them is that a warning text is displayed in the demo version

[8]See PanoTools in Section II for a list of panoramic viewers.
[9]http://www.fraps.com/

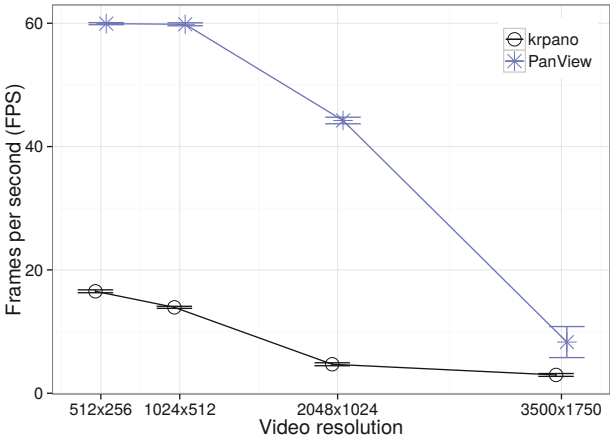


Fig. 6: Frames per seconds (FPS) when playing videos of different sizes considering krpano (panoramic viewer based on Adobe Flash Player) and PanView (based on WebGL and HTML5). Vertical bars indicate a 95% confidence interval of the FPS when playing the video. PanView presents higher FPS than krpano for any video size.

when playing the video. We use a panoramic video with 180 frames scaled to the sizes 512x256, 1024x512, 2048x1024 and 3500x1750. All tests were performed in a computer with Google Chrome web-browser Version 35, Windows 8.1 operational system, processor Intel(R) Core(TM) i7-3537U and 8 gigabytes of RAM.

Results presented in Figure 6 stand that, for any video resolution, PanView presents higher FPS than krpano. We further perform a paired t-test to certify that the differences between the averaged FPS for each video size is significant, which returns a p-value equal to 0.03, asserting that our proposed panoramic viewer presents higher FPS. The results are expected since the WebGL, used to draw videos in the PanView, take advantage of optimized codes for graphical cards in modern web-browsers.

V. Applications

PanView was originally developed to provide a web-based panoramic viewer for two projects under development, which are Historical Town Navigation System, described in Section V-A, and Railroad Analysis, described in Section V-B. Those projects are examples of applications for PanView, since our panoramic viewer extended to an independent project and it can be applied in other environments.

A. Historical Town Navigation System

PanView was originally part of the project Historical Town Navigation System [13], in which the streets of four historical cities of Minas Gerais, Brazil (Ouro Preto, São João Del Rei, Congonhas do Campo and Tiradentes) were registered in panoramic videos. The idea of the project is to provide a virtual city tour using a desktop or a web interface. PanView is the tool developed for the web interface.

B. Railroad Analysis

The idea of this project [14] is to capture panoramic videos from a railroad network for latter analysis. With the panoramic videos, the technicians in transportation engineering can analyze the railroads for events of interest, which are then used to elaborate a revitalization plan. The PanView allows easy access to the panoramic videos, since users interested in developing the revitalization plan can check the events of interest online and update them when convenient.

VI. Conclusions and Future Works

We presented PanView, a panoramic video viewer based on modern web standards, such as HTML5 and WebGL. Our player was built to be easily customizable and to support user modules which enrich the functionalities already presented in the core. We implemented two modules, one to handle a map synchronized with the video and another that provides on-screen buttons and keyboard controls. PanView achieved a better performance compared with an available panoramic viewer based on Adobe Flash Player.

The main advantages presented for PanView were open source code, low computational resource requirements, easy customization, extension modules and no dependence on external plugins. To illustrate the applicability of PanView, we presented two projects: a virtual tour on historical cities, within the Historical Town Navigation System project, and the analysis of a railroad network.

In future works, we will study the inclusion of other panoramic projections in PanView core. The user interaction can be enhanced through the implementation of clickable areas inside the videos, known as hotspots. In order to improve the performance, PanView can be modified to decode only the part of the video frame which is being shown.

Acknowledgment

The authors are grateful to FAPEMIG, CAPES and CNPq funding agencies.

References

[1] D. Anguelov, C. Dulong, D. Filip, C. Frueh, S. Lafon, R. Lyon, A. Ogale, L. Vincent, and J. Weaver, "Google street view: Capturing the world at street level," *Computer*, vol. 43, 2010.

[2] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *International journal of computer vision*, vol. 74, no. 1, pp. 59–73, 2007.

[3] Y. Xiong and K. Pulli, "Fast panorama stitching for high-quality panoramic images on mobile phones," *Consumer Electronics, IEEE Transactions on*, vol. 56, no. 2, pp. 298–306, 2010.

[4] A. Au and J. Liang, "Ztitch: A mobile phone application for immersive panorama creation, navigation, and social sharing," in *Multimedia Signal Processing (MMSP), 2012 IEEE 14th International Workshop*. IEEE, 2012, pp. 13–18.

[5] J. Foote and D. Kimber, "Flycam: Practical panoramic video and automatic camera control," in *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference*, vol. 3. IEEE, 2000, pp. 1419–1422.

[6] R. Pea, M. Mills, J. Rosen, K. Dauber, W. Effelsberg, and E. Hoffert, "The diver project: Interactive digital video repurposing," *MultiMedia, IEEE*, vol. 11, no. 1, pp. 54–61, 2004.

[7] B. Erol and Y. Li, "An overview of technologies for e-meeting and e-lecture," in *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*. IEEE, 2005, pp. 6–pp.

[8] M. Meilland, A. I. Comport, and P. Rives, "Dense visual mapping of large scale environments for real-time localisation," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference*. IEEE, 2011, pp. 4242–4248.

[9] J. Kopf, B. Chen, R. Szeliski, and M. Cohen, "Street slide: browsing street level imagery," *ACM Transactions on Graphics (TOG)*, vol. 29, no. 4, p. 96, 2010.

[10] R. Patil, P. E. Rybski, T. Kanade, and M. M. Veloso, "People detection and tracking in high resolution panoramic video mosaic," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 2. IEEE, 2004, pp. 1323–1328.

[11] R. Kaiser, M. Thaler, A. Kriechbaum, H. Fassold, W. Bailer, and J. Rosner, "Real-time person tracking in high-resolution panoramic video for automated broadcast production," in *Visual Media Production (CVMP), 2011 Conference for*. IEEE, 2011, pp. 21–29.

[12] B. P. Douglass, *Real-time design patterns: robust scalable architecture for real-time systems*. Addison-Wesley Professional, 2003, vol. 1.

[13] M. de Miranda Coelho, "Recuperação de informação visual em bases de imagens de cidades históricas: contribuições para o reconhecimento e classificação de imagens," Doctoral thesis in Computer Science, Universidade Federal de Minas Gerais (UFMG), 2013.

[14] M. F. Porto, L. C. d. J. Miranda, N. T. R. Nunes, and C. E. Santos Jr., "The Feasibility Study for Establishment of Passenger Rail in The Metropolitan Region of Belo Horizonte / Brazil," *5th International Multi-Conference on Complexity, Informatics and Cybernetics (IMCIC)*, pp. 201–206, 2014.

Visualizing Contextual Information in Aggregated Web Content Repositories

Arno Scharl, Ruslan Kamolov, Daniel Fischl, Walter Rafelsberger, Alistair Jones
Department of New Media Technology
MODUL University Vienna
Vienna, Austria
scharl@modul.ac.at

***Abstract*—Understanding stakeholder perceptions and the impact of campaigns are key insights for communication experts and policy makers. A structured analysis of Web content can help answer these questions, particularly if this analysis involves the ability to extract, disambiguate and visualize contextual information. After summarizing methods used for acquiring and annotating Web content repositories, we present visualization techniques to explore the lexical, geospatial and relational context of entities in these repositories. The examples stem from the *Media Watch on Climate Change,* a publicly available Web portal that aggregates environmental resources from various online sources.**

Keywords—Web intelligence; context; visual analytics; word tree; named entity detection; relation extraction; climate change.

I. Introduction

Media analytics solutions have been developed for various domains including sports [6], politics [2; 9] and climate change [4; 8], often focusing on specific aspects such as (sub-)event detection [1], content classification [4] and the automated annotation of video broadcasts [2]. Such media analytics systems face two major challenges:

- compile and annotate large document collections from online sources that are heterogeneous in terms of authorship, formatting, style (e.g., news article versus tweets), and update frequency;
- provide an interactive dashboard to select the most relevant subsets of the information space, and to analyze and visualize the extracted information.

Contextual information, especially when properly disambiguated, plays a vital part in addressing both challenges. It improves several steps in the processing pipelines of media analytics platforms – targeted content acquisition via focused crawling [5], for example, or more accurate knowledge extracting algorithms tailored to the specifics of user-generated content [10] – especially when trying to understand the role of affective knowledge in the decision-making process [3].

II. Media Watch on Climate Change

The *Media Watch on Climate Change* (MWCC) is a content aggregation and online collaboration platform publicly available at www.ecoresearch.net/climate [4; 8]. Using the Web intelligence and media analytics platform of webLyzard (www.weblyzard.com), it compiles large archives of Web content from multiple online sources, and provides a variety of knowledge co-creation and visualization tools [8]. MWCC also serves as the knowledge repository for DecarboNet, a three-year research project funded by the European Commission via the *7th Framework Program* (www.decarbonet.eu).

MWCC integrates multilingual content from English, French and German online sources: social media including *Twitter, Facebook, Google+* and *YouTube,* and the Web sites of news channels, Fortune 1000 companies, municipalities, and environmental NGOs. Automated document enrichment services then transform the gathered information into a contextualized information space spanning geospatial, temporal and social dimensions.

Analyzing this information space sheds light on stakeholder perceptions, reveals flows of relevant information, and provides indicators for assessing the impact of large environmental campaigns such as the *Earth Hour* [11].

III. Web Crawling and Preprocessing

To process and enrich data from unstructured, structured and social evidence sources, MWCC pursues a focused crawling strategy. Managing the abundant quantity and dynamic nature of news and social media content requires efficient pre-processing to remove irrelevant content at an early stage of the processing pipeline. This filtering reduces the number of documents to be processed by computationally expensive information extraction algorithms.

MWCC relies on a domain specificity measure based on a combination of blacklists and whitelists to assess the relevance of gathered documents in the context of climate change and related environmental issues (see Figure 1).

Edit Term List for Topic: Solar

```
solar energy
solar power
solar plant(s)?
photovoltaic(s)?
solar panel(s)?
solar collector(s)?
solar thermal
```

Figure 1. List of regular expressions to define the topic "solar energy"

IV. Extracting Factual and Affective Knowledge

Which organizations tend to have a negative reputation among social media users? Who are the most visible climate change activists, and what are mainstream media associating with their recent public appearances?

978-1-4799-6953-1/14 $31.00 © 2014 IEEE
DOI 10.1109/LAWeb.2014.18

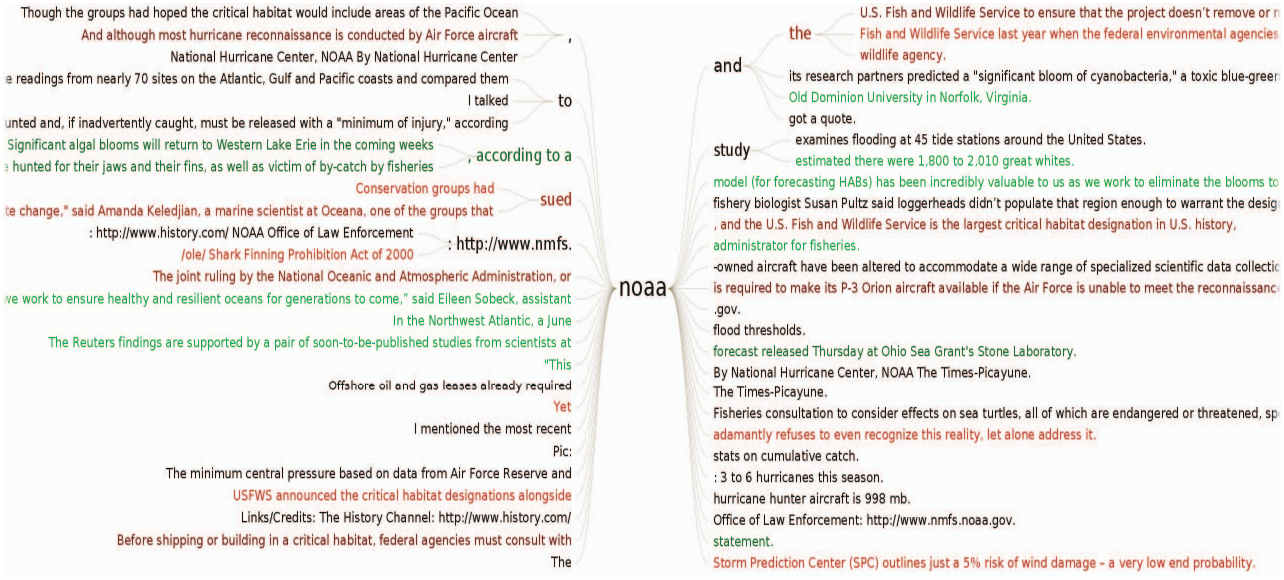


Figure 2. Word tree representation for the search term "NOAA" in Anglo-American news media coverage (Q2/2014)

For properly answering such communication questions, MWCC utilizes *Recognyze* [15] – a named entity recognition and resolution component that draws upon structured external knowledge repositories such as *DBpedia.org*, *Freebase.com* and *GeoNames.org* to identify and disambiguate named entities (organizations, persons and locations), assigning confidence values to align them with the items contained in the external knowledge repositories.

The result is a continuously evolving knowledge repository that helps to better understand networks and the dynamic relations [1] among their actors.

The platform provides a seamless integration of factual (concepts, instances, relations) and affective (beliefs, opinions, arguments, etc.) knowledge:

- *Factual Knowledge.* The Recognyze named entity recognition and resolution component not only identifies and classifies entities, but also grounds them to external knowledge bases or corporate databases.
- *Affective Knowledge* includes sentiment and other emotions expressed in a document, which are captured and evaluated by opinion mining algorithms [13; 14].

V. Visualizing Contextual Information

The MWCC visual dashboard [8] reveals popular issues that are being discussed in conjunction with a given topic. This section describes three new visualization components to reveal contextual information in such online discussions – a *word tree* for lexical context, a *map projection* for geospatial context, and an *entity tracker* for relational context across organizations, persons and locations.

The color coding of the diagrams reflects normalized document sentiment, ranging from green (positive) to grey (neutral) and red (negative).

A. Lexical Context

Once a user has entered a search query, MWCC ranks the matching results by relevance, date, or geographic location. Sentiment information is available in a separate column. Clicking on a quote extends the entry; a second click activates the full text mode. When the full text of a document is shown, the header of the page includes document keywords and the source URL, while the footer summarizes the document's other annotations including source category, source location, target location, sentiment, and relevance.

Alternatively, the system lists matching quotes as a *concordance list*. Users can sort the concordances by their source, date of publication, and sentiment on either a document or sentence level.

The *word tree* module presents the concordance list in a visual and more intuitive manner, summarizing the different contexts in which certain entities or topics are being discussed. Its graph-based display facilitates the rapid exploration of search results and conveys a better understanding of how language is being used surrounding a topic of interest.

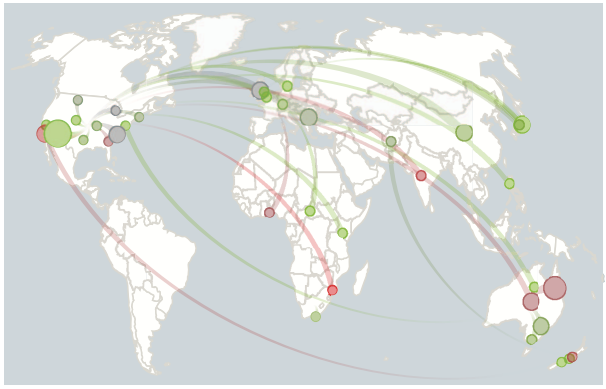
Based on the popular keyword-in-context technique [12], our specific implementation of the *word tree* metaphor adopts a symmetrical approach [7]. The root of the tree is the search term. The left part of the tree displays all sentence parts that occur before the search term (prefix tree), and the right part those that follow the search term (suffix tree). These branches to the left and to the right help users to spot repetition in contextual phrases that precede or follow the search term.

Mouse-over highlights connected elements, allowing users to reconstruct entire sentences. Visual cues include different font sizes to indicate the frequency of phrases, and connecting lines to highlight typical sentence structures.

Figure 2 shows how the tree-like structure is built after searching for the term "NOAA", which is the acronym of the *National Oceanic and Atmospheric Administration*, and grouping identical phrases containing the term into nodes (e.g., "NOAA study"). This grouping together of equal phrases into a connected tree structure sheds light on word usage within the selected source(s) in a given time interval.

B. Geospatial Context

The results of searches within the MWCC portal are also projected onto a geographic map that shows the regional distribution of Web coverage – e.g., references to locations co-occurring with the term "solar energy" as shown in Figure 3. The position of circles is determined by the geographic coordinates of these references, their size is proportional to the number of documents referring to a specific position.



Documents | Quotes | Word Tree | Locations | Sources | Source Map

Location	Count	Latitude	Longitude	Sentiment
California california | san francisco chronicle | solarcity	86	37.3	-119.8	+0.3
Denver proctor | cathy proctor | cathy	51	39.7	-105.0	+0.3
Washington american coalition | asthma | avalanche	43	47.5	-120.5	+0.2
New York solarcity | rive | musk	37	43.0	-75.5	+0.3
People's Republic of China solar | solarcity | commerce	36	35.0	105.0	+0.1
Arizona arizona | public service co | corporation commission	32	34.5	-111.5	+0.3
Idaho solar roadways | roadways | brusaw	24	44.5	-114.3	+0.3
North Carolina solar | close proximity | apple	24	35.5	-80.0	+0.5
Colorado xcel | xcel energy | cathy	21	39.0	-105.5	+0.5
Europe detailed analysis | decc | britain	21	48.7	9.1	+0.2
United States pbn | clean energy | address	20	38.9	-77.0	+0.5
Republic of India india | rajasthan | solar	19	20.0	77.0	+0.2
Florida hacking | alibaba | chinese	18	28.8	-82.5	+0.1
Hawaii pbn | shimogawa | hawaii public	15	20.8	-156.5	+0.6

Figure 3. Geographic map and list of locations that co-occur with the term "solar energy" in Anglo-American news media coverage (06-07/2014)

When rendering documents in their geospatial context, the system distinguishes between source and target information – i.e., the authors' locations versus the primary locations referenced in the documents, which is determined by applying the above mentioned *Recognyze* component to a geo-tagging process (the table underneath the map shows a list of the identified geographic entities, sorted by decreasing co-occurrence frequency with "solar energy").

C. Relational Context

To identify opinion leaders and reveal key factors influencing social conversations about a topic, the webLyzard platform detects not only locations, but also other named entities such as persons and organizations that have an impact on news and social media coverage. To develop a deeper understanding of this process, analysts must not only understand how these entities influence topics of interest, but also unravel the interconnected relations among the entities themselves. How did a public appearance of the CEO impact a company's perceived relation to main competitor, for example, and what are journalists associating with the competitor's latest announcement?

To help answer such questions, the *Entity Map* shown in Figure 4 visualizes (i) relations among named entities in the analyzed corpus, and (ii) co-occurrence patterns between these entities and user-defined search terms. In the case of MWCC coverage from April to July 2014, the list of referenced entities includes politicians such as U.S. President *Barack Obama* and the Australian Prime Minister *Tony Abbott*, organizations such as the *Green Energy Collective*, and various locations including *Washington DC* and the *State of California*. The Entity Map component combines a line chart with a radial imposition, and a radial convergence diagram:

- *Radial Convergence Diagram.* Located in the center of the graph, the radial convergence diagram displays relations among different entities using ribbons. Entity names are displayed along a circle – their font size indicates the number of documents that mention the entity, their color ranges from red to green depending on the average sentiment (in line with the sentiment color coding of the word tree and the geographic map). The thickness of an arc represents the number of co-occurrences between an entity pair. On mouse-over, the opacity of arcs that connect the selected entity to other entities is increased. A slider element in the lower left corner controls the level of detail in the radial convergence diagram – i.e. it determines the threshold for showing relations among entities. The second slider element in the lower left corner adjusts the number of entities to be shown, between a minimum of three and a maximum of 50 entities.
- *Line Chart.* Surrounding the radial convergence diagram in the center, the data points in the line chart show the number of co-occurrences between an entity and the selected topics (using the same color-coding as the trend chart). To increase the readability of the display and facilitate comparisons across topics, the line chart uses a logarithmic scale.

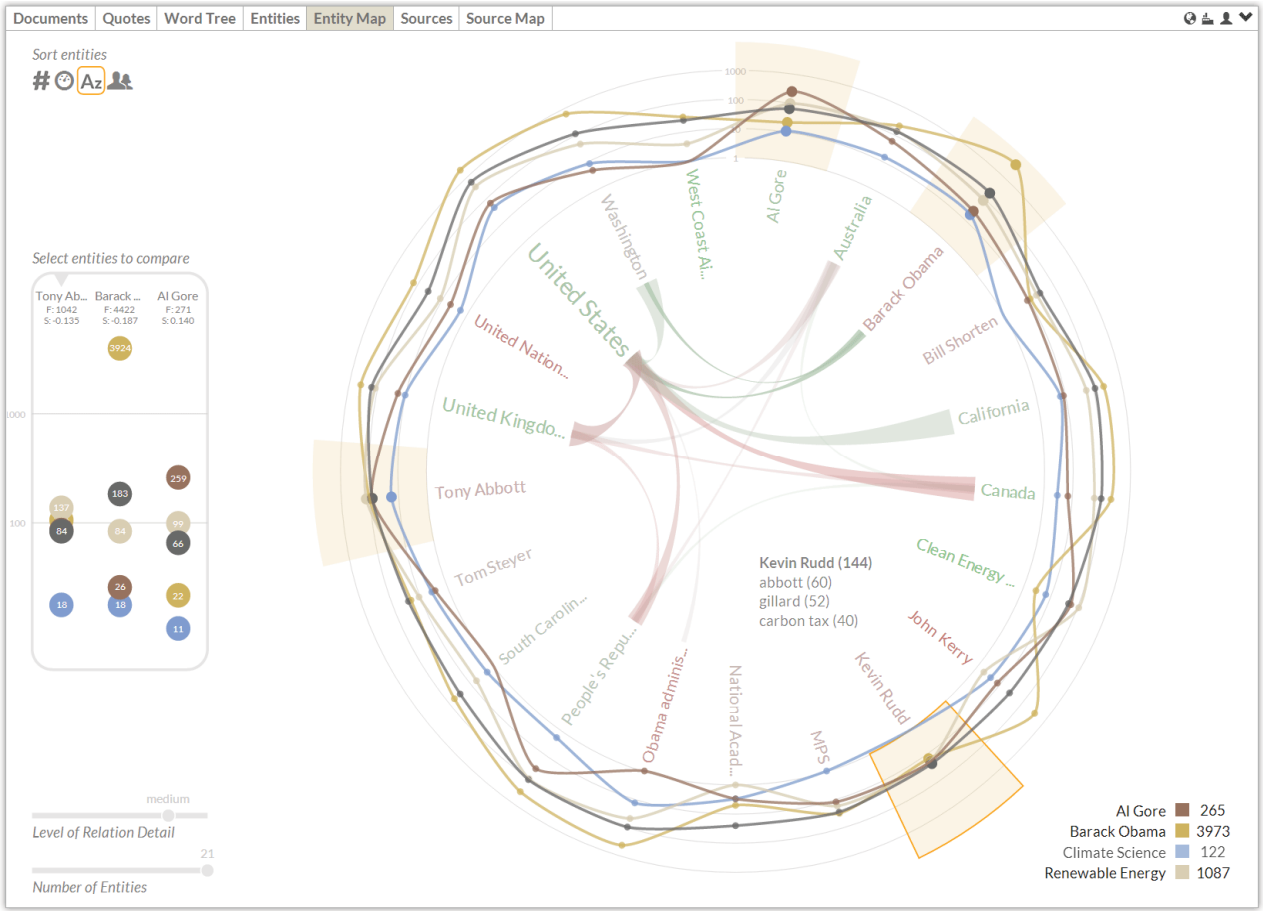


Figure 4. Entity Map showing the co-occurrence frequency of the search terms "Al Gore", "Brarack Obama, ", "Climate Science" and "Renewable Energy" with identified named entities (organizations, persons and locations), as well as the strength of the relations among these entities

Three small icons in the upper right corner control which entity types are displayed – persons, organizations and locations (of which at least one needs to be active). Clicking on an icon in the upper left corner causes the entities to be rearranged by (i) entity type, (ii) name, (iii) the number of documents which contain an instance of the entity's name in descending order, or (iv) the average sentiment of the documents containing the entity, from positive to negative.

Both the line chart and the radial convergence diagram are being updated by means of smooth, animated transitions. Hovering over an entity highlights the corresponding sector, shows a tooltip with the top three keywords associated with the chosen entity, and highlights the arcs in the radial convergence diagram. Hovering over one of the search terms in the list on the left removes all lines in the chart except for the one corresponding to the selected search term; this allows for a cleaner view of a single search term.

Additional interactions support more detailed comparisons. Clicking on an entity causes supplemental information to be displayed in a sidebar, which includes the data points with the co-occurrence values and the entity information – i.e., name, document count (d) and average sentiment (s).

The logarithmic scale of the sidebar adjusts automatically to accommodate the range of data values. The box contains the three most recently selected entities, which remain highlighted in the graph.

VI. Summary and Conclusion

The visualizations presented in this paper allow users to interactively explore the lexical, geospatial and relational context of Web documents. The underlying data stems from the *Media Watch on Climate Change*, a media analytics portal available at www.ecoresearch.net/climate. The system is currently being extended into a collective awareness platform through the *DecarboNet* project (www.decarbonet.eu). The context of Web coverage is important when aiming to investigate and better understand the various processes that lead to collective awareness, since it impacts opinions and decision making on both individual and collective levels.

The tools presented in this paper help to understand the context of Web coverage by establishing connections between named entities (persons, organizations, and locations), based on references to these entities in aggregated content from English, French and German news channels, and from

social media platforms such as Twitter, Facebook, Google+ and YouTube. The *Media Watch on Climate Change* utilizes the *Recogynze* component (www.weblyzard.com/recognyze) to identify and resolve named entities. *Recognyze* draws upon structured external knowledge repositories such as *DBpedia.org*, *Freebase.com* and *GeoNames.org* to disambiguate these entities via confidence values that align entities with the items of the knowledge repositories.

Extracting and visualizing contextual information transforms unstructured collections of crawled Web content into structured repositories of actionable knowledge. Thereby, the presented techniques to reveal context in Web coverage provide value for a wide range of organizations including enterprises, non-government entities, news media outlets, science agencies, and policy makers. Uncovering patterns and trends in Web coverage can help these organizations to adopt better strategies for engaging audiences, guide their communication and public outreach campaigns, and increase the effectiveness of their decision making processes.

ACKNOWLEDGEMENT

The research presented in this paper has been conducted as part of the DecarboNet project (www.decarbonet.eu), which has received funding by the European Union's 7th Framework Program for research, technology development and demonstration under the Grant Agreement No. 610829.

REFERENCES

[1] Adams, B., Phung, D. and Venkatesh, S. (2011). Eventscapes: Visualizing Events Over Times with Emotive Facets. 19th ACM International Conference on Multimedia (MM-2011). Scottsdale, USA: 1477-1480.

[2] Diakopoulos, N., Naaman, M. and Kivran-Swaine, F. (2010). Diamonds in the Rough: Social Media Visual Analytics for Journalistic Inquiry. IEEE Symposium on Visual Analytics Science and Technology (VAST-2010). Salt Lake City, USA: IEEE: 115-122

[3] Hoang, T.-A., Cohen, W.W., et al. (2013). Politics, Sharing and Emotion in Microblogs. IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. Niagara Falls, Canada: ACM Press: 282-289.

[4] Hubmann-Haidvogel, A., Scharl, A. and Weichselbraun, A. (2009). "Multiple Coordinated Views for Searching and Navigating Web Content Repositories", Information Sciences, 179(12): 1813-1821.

[5] Mangaravite, V., Assis, G.T.d. and Ferreira, A.A. (2012). Improving the Efficiency of a Genre-aware Approach to Focused Crawling Based on Link Context. Eighth Latin American Web Congress (LA-WEB 2012). Cartagena de Indias, Colombia: IEEE CPS: 17-23.

[6] Marcus, A., Bernstein, M.S., et al. (2011). Twitinfo: Aggregating and Visualizing Microblogs for Event Exploration. 2011 Annual Conference on Human Factors in Computing Systems (CHI-11). Vancouver, Canada: ACM: 227-236.

[7] Muralidharan, A., Hearst, M.A. and Fan, C. (2013). WordSeer: A Knowledge Synthesis Environment for Textual Data. 22nd ACM International Conference Information and Knowledge Management (CIKM-2013). San Francisco, USA: ACM: 2533-2536.

[8] Scharl, A., Hubmann-Haidvogel, A., et al. (2013). "From Web Intelligence to Knowledge Co-Creation – A Platform to Analyze and Support Stakeholder Communication", IEEE Internet Computing, 17(5): 21-29.

[9] Shamma, D.A., Kennedy, L. and Churchill, E.F. (2010). Tweetgeist: Can the Twitter Timeline Reveal the Structure of Broadcast Events? ACM Conference on Computer Supported Cooperative Work (CSCW-2010). Savannah, USA.

[10] Sipos, R., Ghosh, A. and Joachims, T. (2014). Was this Review Helpful to You?: It Depends! Context and Voting Patterns in Online Content. 23rd International World Wide Web Conference (WWW-2014). Seoul, Korea: World Wide Web Consortium: 337-347.

[11] Sison, M.D. (2013). "Creative Strategic Communications: A Case Study of Earth Hour", International Journal of Strategic Communication, 7(4): 227-240.

[12] Wattenberg, M. and Viégas, F.B. (2008). "The Word Tree, an Interactive Visual Concordance", IEEE Transactions on Visualization and Computer Graphics, 14(6): 1221-1228.

[13] Weichselbraun, A., Gindl, S. and Scharl, A. (2013). "Extracting and Grounding Contextualized Sentiment Lexicons", IEEE Intelligent Systems, 28(2): 39-46.

[14] Weichselbraun, A., Gindl, S. and Scharl, A. (2014). "Enriching Semantic Knowledge Bases for Opinion Mining in Big Data Applications", Knowledge-Based Systems: Forthcoming (Accepted 26 Apr 2014).

[15] Weichselbraun, A., Streiff, D. and Scharl, A. (2014). Linked Enterprise Data for Fine Grained Named Entity Linking and Web Intelligence. 4th International Conference on Web Intelligence, Mining and Semantics (WIMS-2014). Thessaloniki, Greece: ACM Press.

A Generation Environment for Front-end Layer in E-government Content Management Systems

Vitor Mesaque Alves de Lima, Ricardo Marcondes Marcacini, Marcelo Henrique Pereira Lima
Campus de Três Lagoas
Universidade Federal de Mato Grosso do Sul
Três Lagoas, Brazil
{vitor.lima, ricardo.marcacini, marcelo.lima}@ufms.br

Maria Istela Cagnin, Marcelo Augusto Santos Turine
Faculdade de Computação
Universidade Federal de Mato Grosso do Sul
Campo Grande, Brazil
{istela, turine}@facom.ufms.br

***Abstract*—The internet is a way by which organizations may show its identity, its purposes, its achievements, providing services and information to the public. A Content Management System (CMS) provides an efficient solution for content managing for websites and portals in general. Researches show that the organizations are interested in cost reduction associated to software development, and it is essential to use automated tools and a reuse systematic process. In this paper we present a generation environment for front-end layer in e-government CMS, in context of systematic reuse using Software Product Line (SPL) automated through frameworks, application generators and reuse repository. The generation environment implements automated mechanisms to reduce accessibility problems in generated web applications.**

Keywords—software product line; application generators; framework; reuse repository; e-government; web accessibility

I. INTRODUCTION

The recent developments in Information Technology and Communication (ICTs) have enabled significant improvements in Electronic Government (e-Gov) services, allowing high social participation in government decisions as well as increasing the government transparency and democracy [12]. In this scenario, one of the important online services offered to citizens are the e-Gov portals, which are Web-based applications where different sectors of public administration provide your identity, planning, actions, and outcomes for society [15]. Therefore, the e-Gov portals are key services for promotion of digital democracy and should ensure universal access to information for all citizens without distinction [12].

Although international standards and guidelines for the accessibility of Web-based applications are widely available, such as the WCAG (Web Content Accessibility Guidelines) [16] and e-MAG (Accessibility Model of Electronic Government) [7], many e-Gov portals still have critical issues that limit universal access to the information. For example, a study published in 2010 showed that 98% of the Brazilians e-Gov portals lack basic requirements for Web accessibility [4], thereby hindering access by people with disabilities. These facts have increased the demand for Web accessible environments in the context of e-Gov management systems.

Besides the aspects related to Web accessibility, government organizations also need an agile management process of the information published on the e-Gov portals. In this sense, the use of the Content Management Systems (CMS) has been a useful solution, since they allow the software reuse and the reduction of effort, cost and time of the system management [8] [13] [14]. However, most existing general-purpose CMS, such as Joomla! [1], Drupal [2], PLONE [3], OpenCMS[4], and Typo3[5], also lack the basic requirements of Web accessibility in the front-end layer [2] [10]. Moreover, a study reported by Lima (2013) [10] showed that even the CMS proposed specifically for e-Gov portals do not adequately address the requirements of Web accessibility [11].

Considering the challenges discussed above, an interesting direction of research in the e-government field is Software Product Lines (SPL) for developing accessible Web solutions. According to Pohl et al. (2005) [13], a SPL can be defined as a set of applications developed using platforms and mass customization. In particular, frameworks and application generators can be applied to SPL automation, thereby optimizing the process of instantiation of Web applications. In this context, Carromeu et al. (2010) [3] proposed a SPL architecture for the generation of e-Gov Web applications, and defined a generation environment, which implements the SPL architecture, composed essentially by Titan Framework tool. An advantage of the Titan Framework is the systematic software reuse through a component-based architecture that implements a generation environment for back-end layer. Moreover, the generation of a new instance (Web application) can be automated by a graphical tool, called Titan Architect, which facilitates the parameter settings of the new instance. Some government Web applications have been developed successfully using the Titan Framework, such as Web portal at the Foundation for Research Support of the state of Mato Grosso do Sul – Fundect, the Federal University of Mato Grosso do Sul – UFMS, the National Phone-in System of

[1] Joomla!: http://joomla.org/
[2] Drupal: http://drupal.org/
[3] Plone: http://plone.org/
[4] OpenCMS: http:// opencms.org/
[5] Typo3: http://typo3.org/

978-1-4799-6953-1/14 $31.00 © 2014 IEEE
DOI 10.1109/LAWeb.2014.20

Accusations – DDN 100 for the Special Secretary of Human Rights from the Presidency of the Republic and the Information System of Evaluating Cultural Projects – SIAC (http://minc.ledes.net/) for the Ministry of Culture.

In this paper we introduce the Titan Frontend, a generation environment for accessible front-end layer in E-government Content Management Systems. Our generation environment extends the Titan Framework (generation environment for the back-end layer) and provides a template engine for accessible e-Gov portals. To support the Titan Frontend, we also proposed a repository for software reuse called Titan Extensions, which manages software components and accessible Web templates for creating new CMSs. The accessibility of Web templates generated for the front-end layer is automatically evaluated by using the ASES (Accessibility Simulator and Evaluator for Sites) model, an evaluator and simulator of the Web accessibility for Brazilian e-Gov portals [1]. For this purpose, the proposed Titan Frontend presents a public Web service, called WebASES, which enables online evaluation of the Web accessibility based on eMAG guidelines, thereby controlling the accessibility of the templates inserted in the repository. To demonstrate the effectiveness of Titan Frontend, we present an evaluation using six e-Gov portals of Brazilian city halls. We carried out a comparative analysis of the current e-Gov portals and new portals generated by the Titan Frontend and the results showed that the front-end layer generated by our SPL process is superior from the perspective of Web accessibility.

The remainder of this paper is organized as follows. Section II presents the details of the generation environment proposed in this paper. Section III presents an evaluation of Titan Frontend as well as discussion of the results. Finally, Section IV presents the conclusions and directions for future work.

II. Generation Environment

The generation environment of the SPL for e-Gov accessible CMSs is divided into two generation layers (back-end and front-end), as shown in Fig. 1. Thus, both layers use application generators and frameworks to automate the instantiation process. The back-end layer is first instantiated, and after it the second front-end layer is instantiated. In practice, the instantiation of the front-end layer results in a public area of an e-Gov portal, on the other hand, the back-end layer is a CMS, where the information is entered, stored and retrieved for display on the front-end layer.

The back-end generation layer, proposed by Carromeu et al. (2010), is composed by two tools: the Titan framework and the Titan Architect application generator. The Architect generates the XMLs (eXtensible Markup Language) and SQLs (Structured Query Language), files necessary to instantiate the Titan Framework, and makes configurations of new instances in graphical form. Thus, the e-Gov portals are developed from a common generic architecture implemented by Titan and a set of reusable artifacts. Therefore, each Web application is an instance of Titan Framework that takes as input XMLs files and transforms it in a CMS at runtime. Its architecture has grey-box, flexible, and extendable characteristics, as well as provides diverse native features such as an authentication system and security log, revision control and authorship.

In turn, the front-end generation layer, which we introduced in this paper, is composed by the Titan FrontEnd application generator and Zend framework[6], such that the Titan FrontEnd automatically generates code for an instance of Zend framework. The Titan Extensions in turn, act on both layers of the environment.

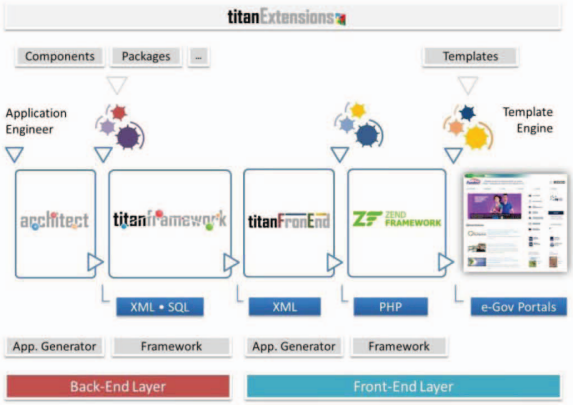


Fig. 1. SPL Environment Generation for e-Gov portals accessible.

A. Titan FrontEnd Tool

Application generators are software systems that transform specifications into an application. Through the use of generators, the application engineer inserts only information about "what" should be done, and the tool decides "how" the information must be transformed into source code [5]. There are two possibilities to build an application generator: constructing a compiler or a composer. Building a compiler means creating a lexical, syntactic and semantic analyzer for a language. Build a composer means creating a software project, derive a set of templates from that project, create a mapping between the specification and these templates, and then use a tool to generate artifacts based on the specifications and templates [17].

The Titan FrontEnd tool is available at http://lives.ufms.br/titanfrontend under the Creative Commons license - (by-nd). The tool can be classified as an application generator, composer, and wizard, able to receive specifications and transform them into software [5]. The Titan FrontEnd composes the environment generation SPL for automated creation and mass customization of CMSs. The tool receives specifications (XML files) provided by a given instance (generated by the back-end generation layer) and generates the source code, at runtime, to the instantiation of the front-end layer. In order to minimize the complexity of the generated code and leverage existing solutions, we chose to use the Zend Web application framework. In summary, the application generator Titan FrontEnd, based on the specifications, automatically generates the source code of the classes that

[6] Zend Framework: http://framework.zend.com

implement the MVC (Model-View-Controller) [9] pattern of the Zend application. Thus, in the generation process, each section of the Titan Framework (back-end layer) was mapped on a Zend module. Fig. 2 shows the resulting directory tree after generation. This automation approach enables a coding process more agile and less susceptible to human error, since the generators can produce safer code than traditional programming methods [5] [6].

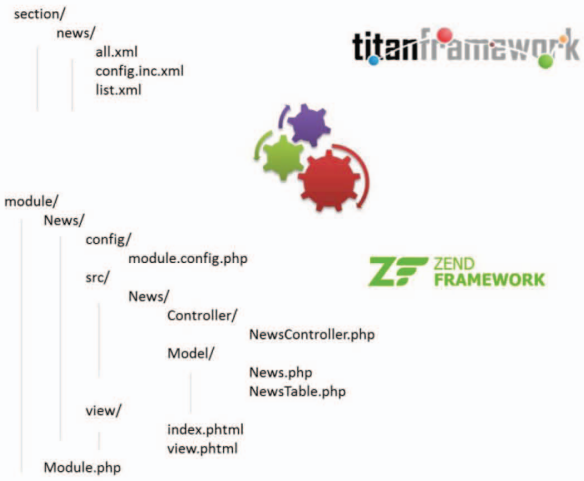


Fig. 2. Directory tree after generation.

The Titan FrontEnd has a template engine that renders the application layout from provided templates. In the generation process, the application engineer can choose predefined layout templates or import it by Titan Extensions repository. Thus, the Web application graphical interface is rendered based on the template layout chosen. Through a well-defined interface, templates can be built and made available on Titan Extensions for reuse, so that the application engineer can change the layout of the pages without the need for coding. Fig. 3 shows an overview of the Titan FrontEnd architecture.

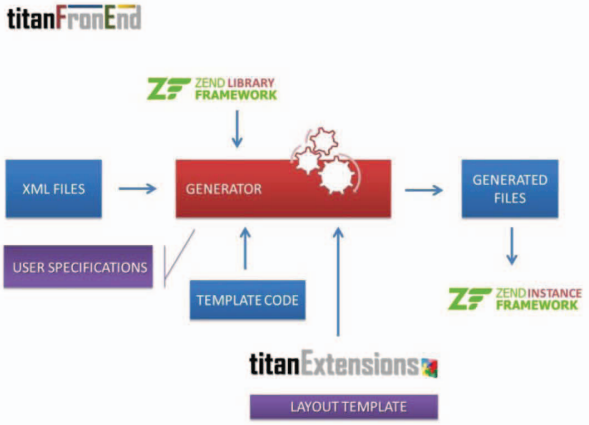


Fig. 3. Titan FrontEnd Architecture.

The generating process is performed in eight steps, as illustrated in Fig. 4, as well as detailed below: i) Zend Configuration: define settings for instantiation of the Zend Framework; ii) Titan Sections: choose the sections that will be generated; iii) Titan Field Sections: choose which attributes/data sections will be generated for the actions list and visualization; iv) Layout Settings: select the layout options; v) Navigation: pagination settings and definition of the content of each section; vi) Advanced Custom: presets positioning of interface elements; vii) Generator: presents a summary of important information relating to the generation and confirmation to the generation process; viii) Finish: presents log of operations performed during the generation process are shown. At the end of the generation process the application engineer can preview the generated portal.

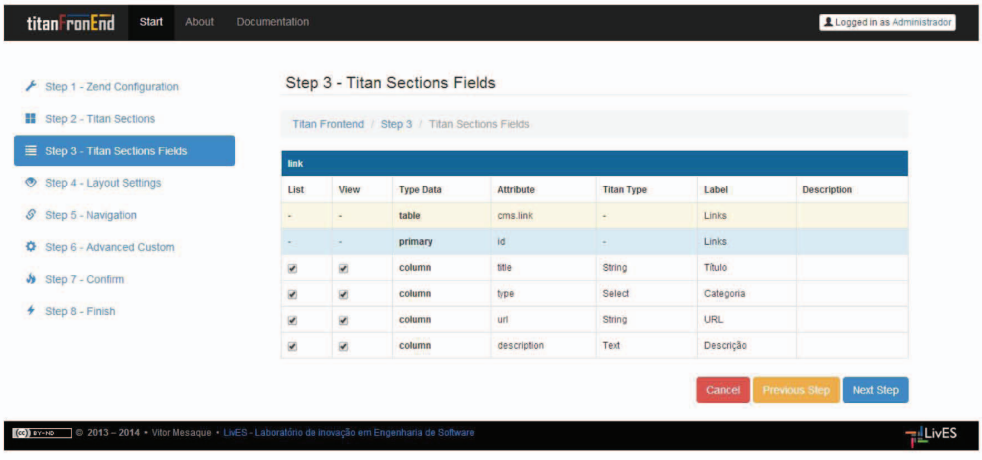


Fig. 4. Step 3: choose which attributes/data sections will be generated for the actions list and visualization.

At each step of the generation process, the checks and validations related to specification generation are performed to provide a safe process. The generation process automates the deployment of some navigational mechanisms such as paging, auto-contrast, resizable font, navigation menu, breadcrumbs, and retrieval system, among others.

B. Accessibility Control Mechanisms

The solution proposed in this paper for the accessibility control in e-Gov portals combines automated mechanisms and the latest Web technologies such as HTML5 (HyperText Markup Language) and CSS3 (Cascading Style Sheets), with the aim to achieve satisfactory levels of accessibility and providing customizable and flexible layouts. The new versions of HTML and CSS provide more semantic and accessible pages and provide more interactivity without the need of installing plugins and loss of performance [15].

The initial code generated by Titan FrontEnd based on the predefined layouts by the generator is accessible and has no error that hinders access to content by people with disabilities. However, the use of custom layout templates may result in insertion of accessibility errors in the code. In this sense, a mechanism for automatic verification of assets was implemented in the Titan Extensions reuse repository to ensure that the available assets in the repository are accessible. Was developed the application called WebASES (available in http://lives.net.br/webases/), which is a Web version of the ASES tool. The WebASES implements a Web service that allows automatic validation of layout templates available in the repository. Thus, whenever a template is added to the repository, its accessibility is evaluated automatically.

III. EVALUATION

In order to evaluate the proposed generation environment, we present an evaluation of six e-Gov portals of Brazilian city halls. Initially, we evaluated the current web accessibility of each front-end layer of the e-Gov portals considering the e-MAG guidelines. The WebASES tool was used to assess the accessibility of the portals. Then, we carried out a simulation for automatic generation of the front-end layer by using the proposed Titan FrontEnd and the templates provided by Titan Extensions. For the evaluation we chose the news section of each portal to perform the comparison. The accessibility errors were categorized according to the priority levels of the e-MAG (Priority 1, Priority 2 and Priority 3).

Below we present the results of the comparative analysis between the current and new portals generated by Titan FrontEnd. In all results, the portals generated by Titan FrontEnd had no accessibility errors classified as Priority 1 (that are points that the creators of Web content should fully satisfy), as shown in Table I. According to the results, all the portals evaluated had accessibility errors of Priority 1, especially the portal of the Belo Horizonte city, with 431 accessibility errors of Priority 1.

TABLE I. COMPARISON OF ACCESSIBILITY ERRORS BETWEEN CURRENT E-GOV PORTALS AND E-GOV PORTALS GENERATED BY TITAN FRONTEND.

Priority Level	Aracaju	Bauru	Belo Horizonte	Campo Grande	Niteroi	São Luis
Accessibility errors of the current e-Gov portals						
Priority 1	83	192	431	76	47	62
Priority 2	17	9	43	12	2	5
Priority 3	0	0	14	0	0	0
Accessibility errors of the e-Gov portals generated by Titan Frontend						
Priority 1	0	0	0	0	0	0
Priority 2	0	0	0	0	0	0
Priority 3	0	0	0	0	0	0

Fig. 5 presents the current e-Gov portal and the new e-Gov portal generated by the Titan Frontend for the Aracaju city, based on developed layout template. It is important to note that the automatically generated front-end layer of the e-Gov portal is similar to the original, without the web accessibility errors.

Fig. 5. Current e-Gov portal and new e-Gov portal generated by the Titan Frontend for the Aracaju city.

IV. CONCLUSIONS

In this paper we presented the Titan Frontend, a generation environment that automates the SPL in domain of e-Gov accessible portals. The tools related to Titan Frontend have been successfully applied in development of new e-Gov portals and have shown that the efforts and resources expended on SPL instantiation process are systematically reused. The Titan FrontEnd presents an automated solution that can significantly reduce the costs associated with the development of front-end layer of e-Gov portals and provide a systematic and reliable process of generation. The reuse and quality of artifacts can be maximized in SPL instantiation with the support of Titan Extension reuse repository. The WebASES tool, in turn, automates the validation process of layout templates to ensure that they are accessible, reducing barriers to accessibility in generated e-Gov portals.

The environment generation and the tools have been constantly validated. Since it was created, some new WebApps were developed and others are currently being developed, such as the Web portal at the Regional Chemistry Council of Mato Grosso do Sul - CRQ-XX (http://crq.lives.net.br) – Ministry of Labor, Web portal at the Observatory of Human Resources in Mato Grosso do Sul (http://observarh.ufms.br/) – Ministry of Health and Web portal at the Week of Education (http://lives.ufms.br/semanadeeducacao) – Ministry of Education.

The generation environment has been constantly developed and case studies in other domains have been carried out. Moreover, new implementations, for instance for a mobile platform, show its considerable flexibility. Research are being conducted in order to identify and map new navigational features that can be automatically generated and incorporated into the Titan FrontEnd instances as well as the exploration of new technologies to improve the generation of web accessible e-Gov portals.

REFERENCES

[1] ASES. Accessibility Simulator and Evaluator for Sites, http://www.governoeletronico.gov.br/acoes-e-projetos/e-MAG/ases-avaliador-e-simulador-de-acessibilidade-sitios, 2012. (In Portuguese)

[2] Burzagli, L., Gabbanini, F., Natalini, M., Palchetti, E., Agostini, A. Using Web Content Management Systems for Accessibility: The experience of a Research Institute Portal. Lecture Notes in Computer Science, v. 5105 LNCS, p. 454-461, 2008.

[3] Carromeu, C., Paiva, D.M.B., Cagnin, M.I., Rubinsztejn, H.K.S., Turine, M.A.S., Breitman, K. Component-Based Architecture for e-Gov Web Systems Development. Engineering of Computer Based Systems (ECBS), 17th IEEE International Conference and Workshops, p.379-385, 2010.

[4] CGI.br e NIC.br. Dimensions and characteristics of the Brazilian web: a study of gov.br, http://www.cgi.br/publicacoes/pesquisas/govbr/cgibrnicbr-censoweb-govbr-2010.pdf, 2010.

[5] Cleaveland, J. C. Program Generators with XML and Java. Prentice Hall, 2001.

[6] Czarnecki, K. e Eisenercker, U. W. Generative Programming. Addison-Wesley, 2002.

[7] E-MAG. e-MAG: Accessibility Model for Electronic Government, http://www.governoeletronico.gov.br/acoes-e-projetos/e-MAG, 2011. (In Portuguese)

[8] Frakes, W. B. e Kang, K. Software Reuse Research: Status and Future. IEEE Transactions on Software Engineering, v. 31, n.7, p. 529-536, 2005.

[9] Gamma, E., Helm, R., Johnson, R., Vlissides, J. Design Patterns – Elements of Reusable Object-Oriented Software. Addison-Wesley, 1995.

[10] Lima, V. M. A. Software Product Line for e-Gov Portals Accessible. Master's Thesis, Federal University of Mato Grosso do Sul, 2013. (In Portuguese)

[11] López, J. M., Pascual, A., Menduiña, C., Granollers, T. Methodology for Identifying and Solving Accessibility Related Issues in Web Content Management System Environments. Proceedings of the International Cross-Disciplinary Conference on Web Accessibility. Anais...New York, NY, USA: ACM, 2012.

[12] PINHO, J. A. G. Investigating e-government portals of states in Brazil: plenty of technology, little democracy. Public Administration Journal. Rio de Janeiro - RJ. v.42, n. 3, p. 471-93, 2008. (In Portuguese)

[13] Pohl, K., Bockle, G., Linden, F. Software Product Line Engineering: Foundations, Principles, and Techniques. Springer, 2005.

[14] Pressman, R. S. Software Engineering: A Practitioner's Approach. 6th ed. New York, USA: McGraw Hill, 2005.

[15] W3C Brasil. World Wide Web Consortium Escritório Brasil, 2013.

[16] WCAG 2.0. Web Content Accessibility Guidelines (WCAG) 2.0, http://www.w3.org/TR/2008/REC-WCAG20-20081211/, 2012.

[17] Weiss, D. M.; Lai, C. T. R. Software Product-Line Engineering. Addison-Wesley, 1999.