The Algorithmic-Autoregulation essay
a collective and natural focus on self-transparency

RENATO FABBRI
IFSC/USP, Participa.br/SG-PR, labMacambira.sf.net
February 1, 2015

Abstract

There are numerous pursues for a lightweight and systematic account of what is done by a group and containing individuals. The Algorithmic-Autoregulation (AA) is a special case, in which a technical community embraced the challenge of registering their own dedication for sharing processes, self-transparency enhancements, and prove dedication. AA is used since June/2011 by dozens of FLOSS and social developers, with the support of different AA software gadgets and for distinct tasks. Intermittence and activity concentration of users activity follows expected natural properties. Social participation and ontological understandings of AA eases comparative analysis and furthers integration.

Resumo

Keywords: distributed development, FLOSS, social participation, OWL, statistics, anthropological physics

Contents

1 AA concept **3**
1.1 Related work 3
1.2 Historical note 4
1.3 Essay structure 4
2 Design features **4**
3 Use practices **5**
3.1 Words and tags 5
3.2 Messages 6
3.2.1 Shouts 6
3.3 Sessions 6
3.4 Accomplishments 7
3.5 Deviations from AA paradigm 7
4 Software support **8**
4.1 First AA: HTTP server, HTML skin and shell client 8
4.2 AA 0.1 9
4.3 PAAinel 10
4.4 Lalenia interface 10
4.4.1 The #labmacambira@Freenode IRC channel log 10
4.5 Ubiquitous AA 11
5 Data **11**
5.1 The OntologiAA OWL ontology 12
5.2 RDF data 12
5.3 Linkage to external data 12
6 Statistics **15**
6.1 Occurrent activity 15
6.1.1 Time activity 15
6.1.2 User activity 22
6.1.3 Character and token incidence 23
6.1.4 Morphosyntactic incidence 26
6.2 Dependent activity 26
6.2.1 Time and user dependent activity 26
6.2.2 Language and user dependent activity 26
6.2.3 Language and time dependent activity 26
6.2.4 Time-related stability 26
6.3 Network activity 26
6.3.1 Time user networks 26
6.3.2 Lexical user networks 26
6.3.3 Network measures 26
6.3.4 Network primitive sectioning 26
6.4 Principal components formation 26

6.5 Immediate clustering 26
6.5.1 Users clustering 26
6.5.2 Words clustering 26
6.6 Timeslot clustering 26
6.7 Comparative analysis 26
6.8 AA, OCD, and Participa.br 26
7 Results **26**
8 Conclusions **27**
8.1 Further work 27
8.2 Acknowledgments 27

1 AA concept

The Algorithmic Autoregulation (AA) is a self-transparency mechanism for sharing processes, proving dedication, and enhance personal or collective self-transparency. Purposes for AA usage are numerous: enable automated and fair compensation for dedications, ease co-working, introduce newcomers, keeping public historical logs of activities, etc. Indeed, other systems have been designed for such a task (see Section 1.1). A brief characterization of AA is:

- The collective origin, purpose and upkeep. This is a free-culture trait, present within many software, and leads to open software and data as described in Section 4.
- Voluntary logging of messages about ongoing work.
- Enables coordinating distributed team work through individual merit.
- More a practice than a software: AA presents variations on the software support and message composition. Often present features are screencasts, peer validation and periodic messaging.

Transparency in this context should be understood as usual organization or State transparency is: a public account of activities [1]; not directly as transparency in self-knowledge, as is the case in some philosophical and political contexts [2]. One should reach [3] for a noteworthy overview of AA as a Global Software Development (GSD).

1.1 Related work

Authors know of no *civil society transparency* platform. There is a number of transparency initiatives for governments [4], for religious parties [5] and for private institutions [6]. Data analysis methods are derived from Natural Language Processing (NLP) and Complex Networks (CN) fields, constituting a hybrid framework of classical [7, 8] and novel [9, 10] approaches.

1.2 Historical note

7th June, 2013, Cleodon Silva [11] died by heart failure. In his memory, the labMacambira.sf.net group was born (Pedro Macambira was one of this pseudonyms). The AA was conceived as the "cardiac pulse" of the group and is in constant usage since July, 2011. It gathers thousands of messages, tenths of users and hundreds of processes. AA messages present contributions, such as commits to official repositories of Evince, Firefox, OpenOffice, Puredata and other software [3]. A number of other activities were registered: new software elaboration and coding, writing of articles, Wikis and Etherpads; articulation of civil society, academic and state instances; studies and reviews. Even so, AA is highly biased towards software development, as can be observed in Sections 6 and 7, and in the GSD article about AA [3].

1.3 Essay structure

Section 3 describes AA uses incident and envisioned. Section 4 exposes different software written or used for AA. Section 5 is dedicated to data. Section 6 further develops statistics about AA in terms of vocabulary and networks. Section 7 states results and Section 8 a concludes with further works and acknowledgements. Tables and figures are inplace, kept as simple briefings and illustrations. External resources - mainly documents, data and scripts - are referenced for further inspection.

2 Design features

To understand use practices and software support (Sections 3 and 4), one needs to observe core design features of AA:

- Evenly spaced messages should be sent by the AA user. The time lapse is called a "slot" and the message a "shout". A slot might refer to the time lapse and the message, this is context dependent and will be pointed on text if ambiguity occurs.
- Shouts should report the task being tackled and/or a briefing of what was done in the slot.
- Shouts are grouped into "sessions". Each session is ideally linked with a short screencast by the user, with a few dozen seconds of explanation about the AA session.
- Each session is sent by email to a random AA user for validation.

Variants of this features were conceived and practiced. Figure 1 exposes a diagram shared and referenced by AA users in the first months of AA practice.

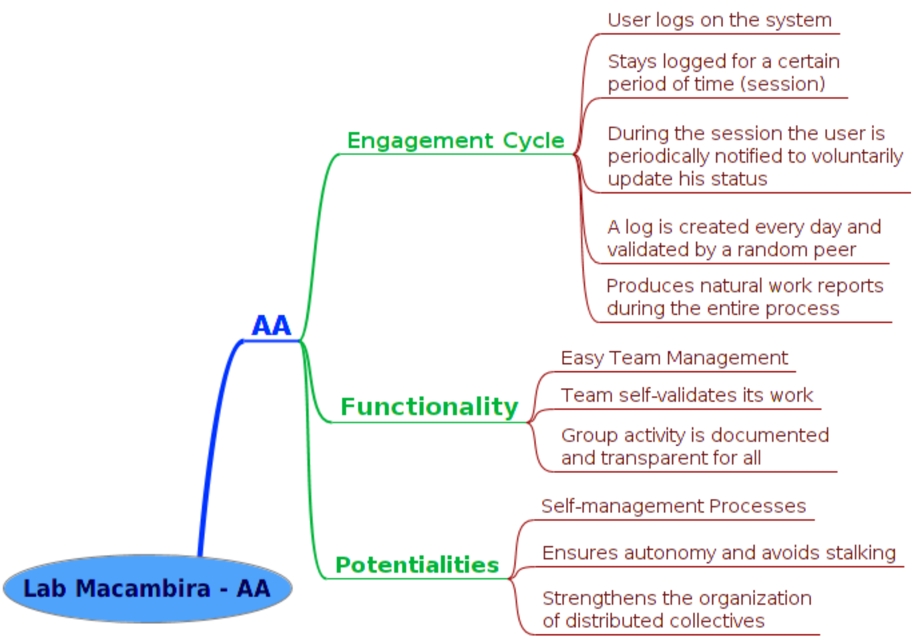

Figure 1: A mind map of the AA methodology shared by users: i) Engagement cycle – the usage of AA; ii) Functionality – the design goals of the system; iii) Potentialities – envisioned benefits of AA by authors of the diagram. As seen in Section 1, core benefits emanate from the self-transparency aspect of AA, with worthy mentions to proving dedications and sharing processes.

3 Use practices

Distinct use methods are incident, mostly regarding the design exposed in Section 2. Even those cases which are not standard can be understood in the light of AA paradigm. Deviations from the ideal case is always present (Section 3.5).

3.1 Words and tags

Throughout AA usage, particular words and tags has been used to classify shouts. Of particular interest are:

- Hashtags, such as #aa, #coding and #articulation. These were inherited from Twitter practice.
- Tags starting with "+" sign, such as +django, +sna and +reading. These aimed at particular used of tagging within AA, with independence

of other systems and easing concurrent use of AA and other social networks.

- Words and abbreviations. Sometimes used in the beginning of shouts, others on the end of them, these also had the purpose of easing categorization of the shouts. These cases sometimes were pointed as tags for entire sessions or for all shouts since tagging, until another tagging shout was sent by the user.

These tagging schemes was also used as a way to enable the "ubiquitous AA", i.e. usage of AA in any social network or communication protocol. The #aao0 tag was used for Twitter streaming AA shouts to a considered database as is the most prominent ubiquitous AA manifestation. Facebook tagging was also used to indicate posts and comments that were AA shouts. On some extreme cases, tagging was used in any platform, considering ubiquitous AA implemented, but not yet mined.

3.2 Messages

Messages for AA usage can be of various types, as shown in Table 4. Usually, the type was dictated by first word of the message. Start messages started an AA session, while stop messages finished an ongoing session. Push messages sent local AA sessions (or independent shouts) to a shared database. There was only one automatic message, designated to register "lost timeslot" of sessions (see Section 3.3). Additional messages were dedicated to query for tickets attributed to the user, milestones and other traditional software development managements facilities.

3.2.1 Shouts

By far the most important AA related message to date is the "shout". Dedicated to expose ongoing tasks, shouts are recurrently envisioned as a structured message, in which the user classifies the shout through special words and tags, and describes ongoing efforts with natural language (usually Portuguese or English). Example of structured shout proposals are in [12] and [13]. Nevertheless, shouts are used by all AA users, in almost all cases, without such sophisticated structure, but as a plain short natural language description of current efforts, i.e. without classification whatsoever of the message.

3.3 Sessions

AA sessions are collections of AA shouts. These have had incidence in AA practice:

- Shouts within a session were input by user each 15 minutes.
- Tolerance for shouts in an ideal session was of ± 5 minutes, considered 15 minutes grid.
- Total duration of 2h, in which 8 shouts should outline tasks and technologies.
- A short screencast was recorded at the end of each session, in which the user exposed dedication within seconds.
- The session was sent to a random user for peer validation in which the session received a score based on shouts and screencast.

Such a session design was very important in first 6 months of AA, where each of almost 10 apprentices were dedicating a session per day. Other users also delivered AA sessions, but not as regularly. Noteworthy: AA shouts can be separated by durations different from 15 minutes: example of incident shout separations include 5 minutes, 2 minutes, 30 minutes, 1h. Most shouts are not explicitly related to sessions, but most shouts still occur in a session-like context. This is regarded as a consequence of the intuitive usage AA sessions were aimed for, and as an inheritance of early AA practice. From now on, AA session is going to be used as meaning both a session registered as such, as an arbitrary time-contiguous set of shouts from the same user. Text will point such specificity if necessary.

3.4 Accomplishments

Processes registered by AA usage often purposes to accomplish something: write a software or an article; make images, music or research scripts; articulate groups, read technical material or take online classes; etc.

These tasks usually spread entire sessions. Sometimes, one session can embrace multiple tasks. A quite common AA usage is to shout one or just a few messages about current efforts, without much care for regularity or completeness.

3.5 Deviations from AA paradigm

There are at least three perceived deviations from AA paradigm, all most often within new users:

- Advertising: shouts containing propaganda about events and groups. This behavior is attributed to both 1) common practice in more commercial platforms, such as Twitter and Facebook; and as 2) the outcome of the AA unusual goals and design, which requires acculturation from a regular visitor before proper understanding.

- Final product exhibitionism: shouts containing not ongoing processes, but only a media or deed recently completed. Although not considered entirely wrong by users, it does not accomplish AA mechanism as posed in Section 1.
- Introduction to AA, IRC and hacking: handling AA is regarded as empowerment and introduction to hacking and open co-working. This first approximation is often marked by playful and test shouts. Although very well esteemed, these messages are also deviations from AA purpose.

4 Software support

Different software support for AA is exposed in this section. Section 5 is dedicated to their integration as linked data, both within AA variants and within participatory instances.

There are mainly three software pieces written to support AA activity. Two of them are a server and client suite each (see Sections 4.1 and 4.2). The third is a fancy dashboard. Among supplementary software support are an automated conversational agents (software [ro]bots), used as alternative User Interfaces (UIs), with a highlight for the Lalenia bot (see Section 4.4); and an initiative to make AA available in all chat networks (see Section 4.5).

All AA software apparatus is contextualized in Table 1.

4.1 First AA: HTTP server, HTML skin and shell client

Although deprecated in favor of AA 01, this first AA software presents the most numerous set of functionalities. Client was completely designed for GNU/Linux terminal usage, and the functionalities are:

- Sending messages to the host.
- Configuration facilities.
- Access to tickets and other software developments facilities.
- Timing to ease AA usage, as described in Sessions 2 and 3.3.

Server functionalities are:

- Receiving shouts and other AA messages through HTTP.
- Registering shouts and other AA messages, received through HTTP, in a MySQL database.

Core HTML skin functionalities are:

- Exhibiting shouts and sessions to other AA users by common WWW HTML pages.
- Interaction of AA users for reviews and screencast attachments to sessions.

Full features of the software extrapolates information above and article scope, as do implementation details. Further information of this and other versions of AA are contextualized in Table 1.

Throughout Jul/2011-Mar/2014, first version of AA, described in this, was used directly of routed from bots (see Section 4.4) and gadgets (Section 4.5).

4.2 AA 0.1

Although there was no online AA software support in the months of April and May 2014, there was AA activity, as seen in Figure 9. This motivated a minimum version of AA to support this visceral usage, that went beyond the software support: AA 0.1 [14].

This implementation targeted the shout message. The dorsal spine is to register shouts independently. All other characteristics should be left to data mining and user dependent tagging.

Minimum client features:

- A simple HTTP call. Integrated trivially as bash commands, to scripts or bots.

Minimum server features:

- Receives the shout with an associated nick and registers to a MongoDB instance with the time the message arrived.
- Returns all shouts as a string or as JSON.
- Heroku Flask app, integrated to online MongoDB. All free online services.

Minimum skin features are part of the server, but listed here for organization:

- Lightest HTML.
- Export as JSON.

Full features of the software extrapolates information above and article scope, as do implementation details. Further information of this and other versions of AA are contextualized in Table 1.

4.3 PAAinel

A fancy skin for visualizing AA activity is pAAinel. Core features are visualization of:

- Latest AA shouts.
- Latest IRC messages.
- Embedded Black Duck Open HUB (former Ohloh) analytics.
- Latest commits to labMacambira.sf.net main repositories.

This Django/Pythonsoftware is written for first AA and has not been adapted to AA 0.1. Full features of the software extrapolates article scope, as do implementation details. Further information of this and other flavors of AA software support are contextualized in Table 1.

4.4 Lalenia interface

To ease usage of AA, and enhance social aspects, the lalenia IRC bot was used as an AA client. This enabled shouts to be logged by IRCusers while on the same channel as lalenia. Core features are:

- Users on the same channel as lalenia can log AA shouts by using the prefix "`;aa `" to a regular message.
- Returns confirmation that AA was logged to AA system. Returns information about AA and software and concepts if successfully logged. Returns an error message if message not logged.
- Both first version and 0.1 have supybot plug-ins.
- A simple Python plug-in [15] for the well known and powerful supybot [16].

4.4.1 The #labmacambira@Freenode IRC channel log

As stated by lalenia, `on #labmacambira there have been 172554 messages, containing 6964778 characters, 1066138 words, 4202 smileys, and 10181 frowns; 178 of those messages were ACTIONs. There have been 31906 joins, 680 parts, 31109 quits, 0 kicks, 4 mode changes, and 133 topic changes.`

Within information in lalenia logs, there were found 1,654 AA shouts that were not in either MySQL or MongoDB databases. Actually, to ease mining, any shout in channel log whose message is identical to any message in all 114,040 messages from MongoDB and MySQL was discarded. Therefore, there was probably a few more shouts in #labmarambira IRC channel log then what is reported here.

4.5 Ubiquitous AA

Following the route posed by AA social network interfaces (see Section 4.4), the Ubiquitous AA is the expansion to all social networks and, indeed, to any media in which activity can be registered. There are two approaches to ubiquitous AA:

1. A software that connects to many messaging services as a bot. One implementation connected to all IRC, Twitter, Google Chat, Facebook, email and MSN. This bot usually receives messages and register them as AA shouts, but can have more elaborate communication procedures [17].
2. Tags with which messages are binded to å activity. This automatically enables AA usage in all social platforms. Messages can be mined for reports and other community usage.

Ubiquitous AA has mythological aspects: is understood as the receiver part of the Yupana Kernel, a mythological entity that receives and spreads all things [18]; and is also understood as a necessary step to human unification [19, 20].

Table 1: All considered AA versions and their databases. References marked with (†) are not currently operational.

version name	**main languages**	**user interface**	**database**	**code repository**	**available at**
First AA(†)	PHP, Python, Bash	linux terminal, HTML	MySQL	[21, 22]	-//-
AA 0.1	Python	linux terminal, HTTP	MongoDB	[14]	[23, 24]
Lalenia bot	Python	IRC	any	[25]	[26]
Ubiquitous AA(†)	Python	IRC, Twitter, Google Chat, Facebook, email, MSN	any	[17]	-//-

5 Data

AA data is scattered among different databases and logs (see Section 4). A coherent integration of these resources is done by means of a dedicated OWL ontology and a mapping routine from relational databases to RDF data.

5.1 The OntologiAA OWL ontology

An ontology, in linked data contexts, is a formalization of a specification. The Web Ontology Language (OWL) is a family of languages designed for authoring ontologies. Core uses ontologies include: 1) reasoning by means of ontological specifications; 2) linking data from different sources; and 3) organization of domain knowledge for coherent consideration. For further considerations of ontologies in a context pertinent to AA, the reader should visit [27, 28].

For AA, an ontology eases key usage aspects:

- Conceptualization of AA is not unique or steady. The ontology delivers a formal paradigm with which community can communicate and develop.
- There is AA related data in different databases, and even in social networks (see Section 4). Common classes to which relate data is a sound method for integrating data.
- AA data is integrated to other social participation instances, such as [29, 30], by super-classes and super-properties of the ontology.

Therefore, the OntologiAAwas delivered by AA community.

5.2 RDF data

The AA data is currently in relational and NoSQL databases, and social networks messages to be mined (Section 4). By using the same ontological background (see OntologiAA in Section 5.1), these data can be translated to RDF for integration and linkage.

Script at [31] outputs RDF from a MySQL database, mostly from first AA version. Script at [32] transcribes a MongoDB database, mostly from AA 0.1, to RDF data. Script at [33] transcribes a shouts from IRC logs, mostly from first AA version, to RDF data.

5.3 Linkage to external data

The usage of RDF and OWL protocols enables linked data facilities. OntologiAA (and thus all AA data) is integrated in two instances of special interest:

- Through OPS, AA data is linked to OPA, OBS, VBS, and OCD. These are Brazilian ontologies, with social participation focus, that has been used for data representation and conceptual studies [28].
- Through all other ontologies (e.g. FOAF, Dublin Core, Schema.org, SIOC), AA data is linked to relevant portions of the Giant Global Graph (GGG) of Linked Open Data (LOD) [34].

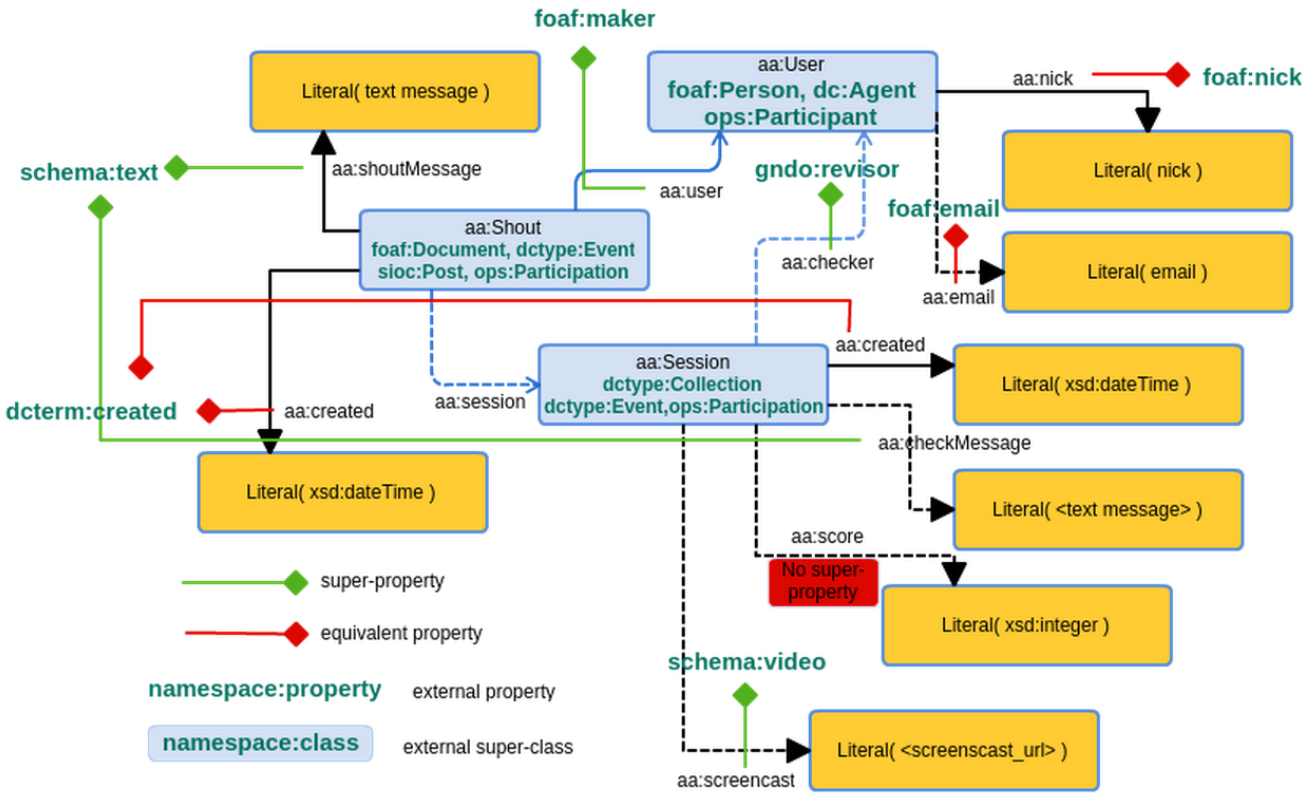

Figure 2: The OntologiAA: an OWL ontology of AA. Classes are concepts related by properties. Classes are also related to upper ontologies (FOAF, Dublin Core, Schema.org, SIOC, GNDO, OPS), as are properties. All properties are functional. Properties with a full line yield existential restrictions to the subjects of the triples suggested by the diagram: `aa:user`, `aa:nick`, `aa:shoutMessage`, `aa:created`. Blue lines mark object properties, while black lines are for data properties. All properties are functional, except for `aa:nick` and `aa:email`. For further information, see Section 5.1.
.

Gives meaning to data while easing data discovery and comparative analysis, to point a few of the advantages of such an approach [35].

Table 2: Registered AAmessages. Operational messages, for signaling session start, stop and publishing local logs (push) are the least abundant. Usage messages with quasi-null semantic content delivers indicative that the user is connected to AA, but no more than that. Messages registering user processes were found to be 34770+1654 = 36424. There were 7504 IRC AA messages, of which 1654 were not registered in databases, probably because of software failures. Automated messages of 'lost timeslot' are the most numerous, with almost half of all messages.

message content	**count**	**type**
push	1718	
start	1169	operational messages = 3936
stop	1049	
empty shouts	92	
empty alerts	83	void messages = 17125
notify	16950	
message shouts	34770	messages about ongoing tasks = 36424
IRC message shouts	1654	
lost timeslot	59863	client automated message
total	115694	all messages are textual

Table 3: Registered AAsessions.

description	**value**
number of sessions	7288
number of shouts in sessions	20299
number sessions with more than 1 shout	905
number of shouts in sessions with more than 1 shout	13916
number of users in sessions	14
number of checkers in sessions	36
number of screencasts in sessions	295
number of scored sessions	191
average session score	3.18
standard deviation of score	0.74
average number of shouts per session	15.38
standard deviation of number of shouts	19.82
first session from	2011-07-06T03:23:05
last session from	2014-04-01T09:11:36

6 Statistics

6.1 Occurrent activity

6.1.1 Time activity

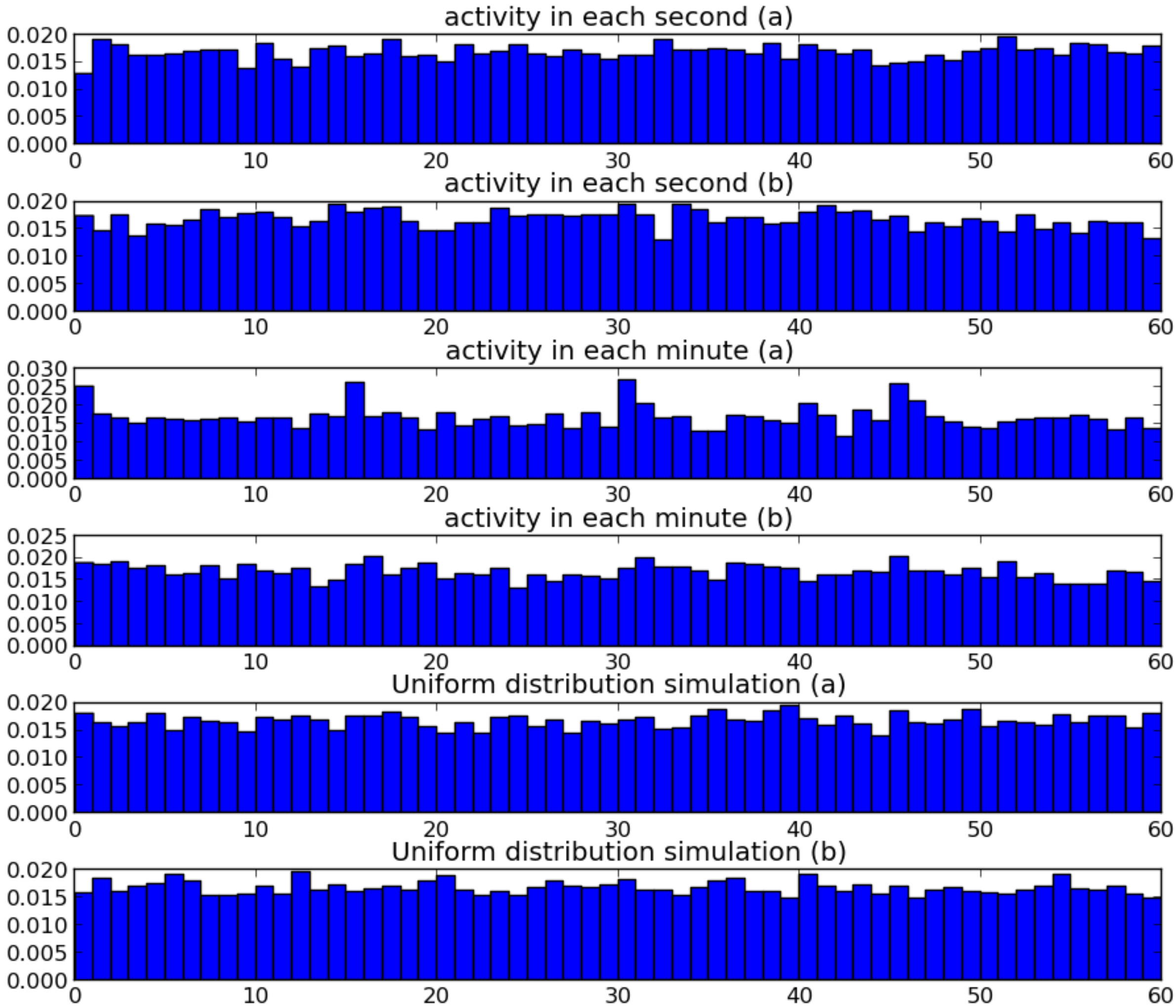

Figure 3: Histogram of AA activity along seconds and minutes. A strong 15 minutes pattern is visible in (a). The same pattern is apparent in all present histograms, although less incisive. This pattern was not found in mailing lists [36], where distribution of activity along seconds and minutes was more homogeneous than Numpy uniform distribution simulator. In AA the scene is the opposite: while simulations delivers $\tau = \frac{max[count(i)]}{min[count(i)]} \approx 1.38$(a) and ≈ 1.26 (b), shouts present $\tau = 1.47$ and $\tau = 1.48$, seconds for (a) and (b), and$\tau = 2.32$ and $\tau = 1.58$, minutes for (a) and (b). Means are considerably bellow 29.5, which might indicate a tendency to shout messages in the beginning of minutes and hours. As these fluctuations among seconds and minutes were not observed in email lists (or other networks, as far as authors know), a hypothesis arises: the tasks at hand and the culture and socioeconomic factors makes timing more prominent. AA itself is time-focused. Set (a) consists of all 15,145 messages from July, 2011 until December, 2011. Set (b) consists of all 21,055 messages from January, 2012 until January, 2015.

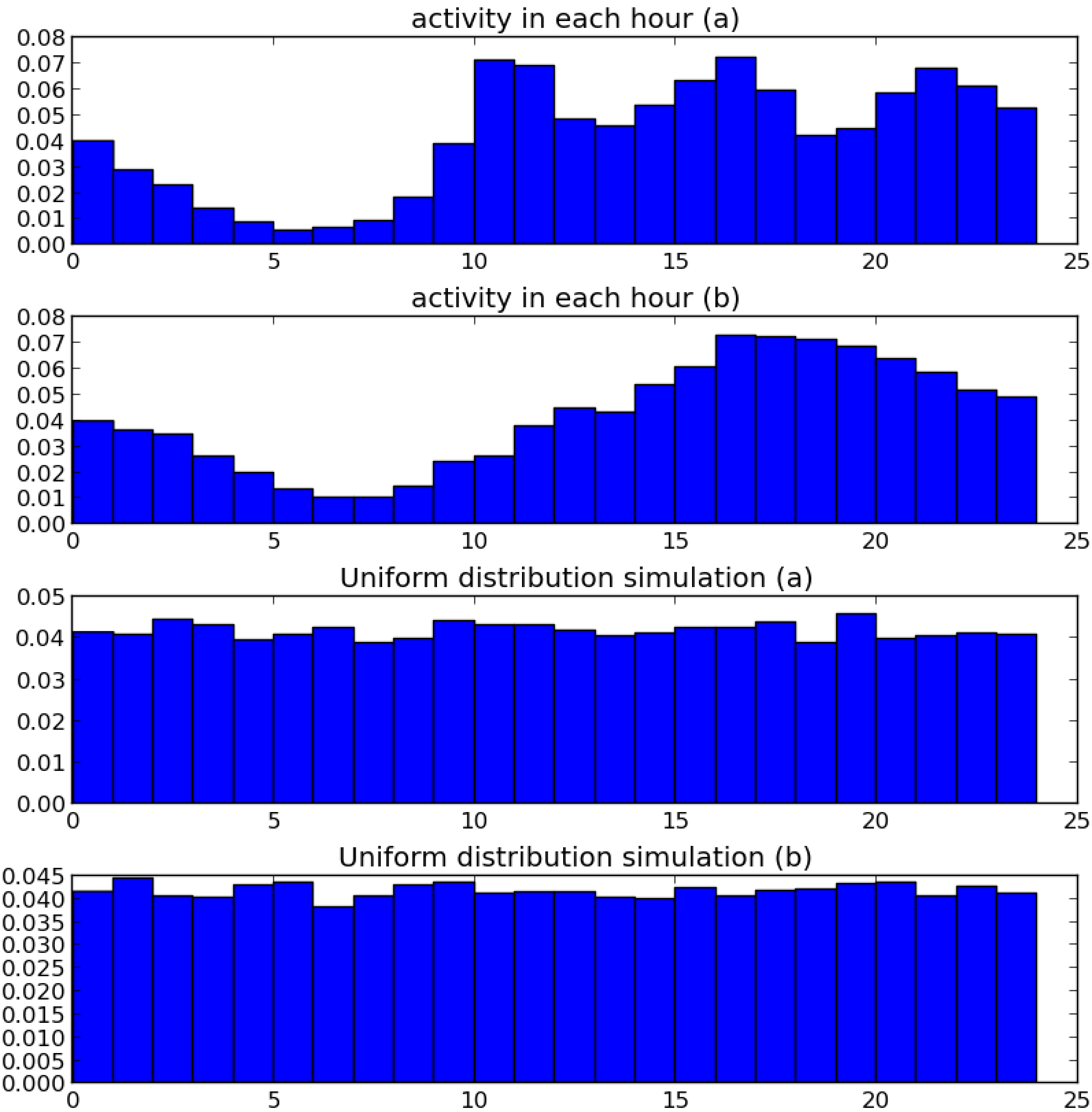

Figure 4: Histogram of AA activity along hours of the day. Discrepancies in extreme incidences are more pronounced than observed with simulation using uniform distributions. While simulations delivers $\tau = \frac{max[count(i)]}{min[count(i)]} \approx 1.18$(a) and ≈ 1.16 (b), shouts present $\tau = 13.5$ and $\tau = 7.16$, respectively. Set (a) consists of all 15,145 messages from July, 2011 until December, 2011. Set (b) consists of all 21,055 messages from January, 2012 until January, 2015. Context (a) is in accordance with results driven from email lists [36], where the activity peak is just before midday, and the second 12h period is the most active. Context (b) exhibits a very odd pattern, in which a climax is reached at 16h in a somewhat smooth progression.

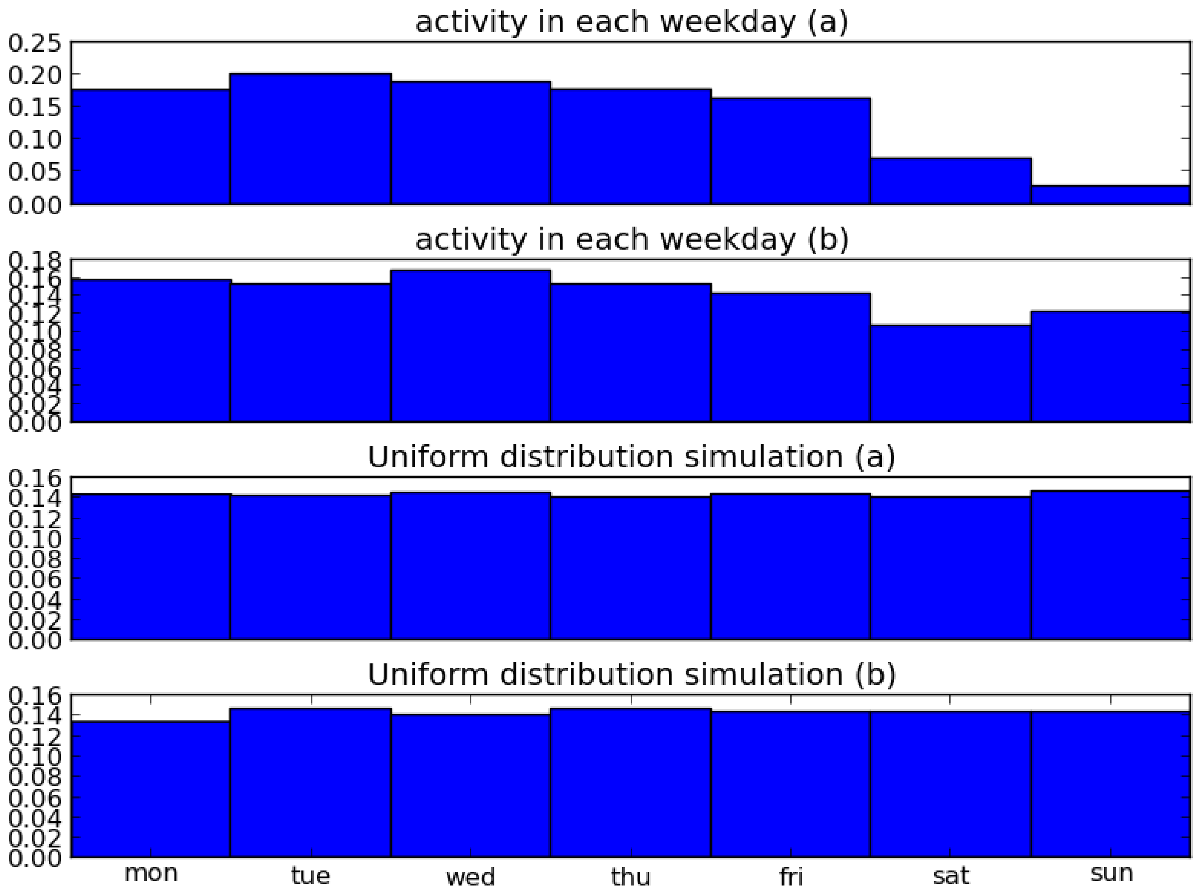

Figure 5: Histogram of AA activity along days of the week. Discrepancies in extreme incidences are more pronounced than observed with simulation using uniform distributions. While simulations delivers $\tau = \frac{max[count(i)]}{min[count(i)]} \approx 1.06$(a) and ≈ 1.07 (b), shouts present $\tau = 7.28$ and $\tau = 1.57$, respectively. Set (a) consists of all 15,145 messages from July, 2011 until December, 2011. Set (b) consists of all 21,055 messages from January, 2012 until January, 2015. Context (a) exhibits a drop of more than half the activity on weekend days. This is more accentuated than results driven from email lists [36], where weekends present about half the activity of other days. Context (b), on the other hand, has a drop of activity on weekends, but maintain levels above 50%.

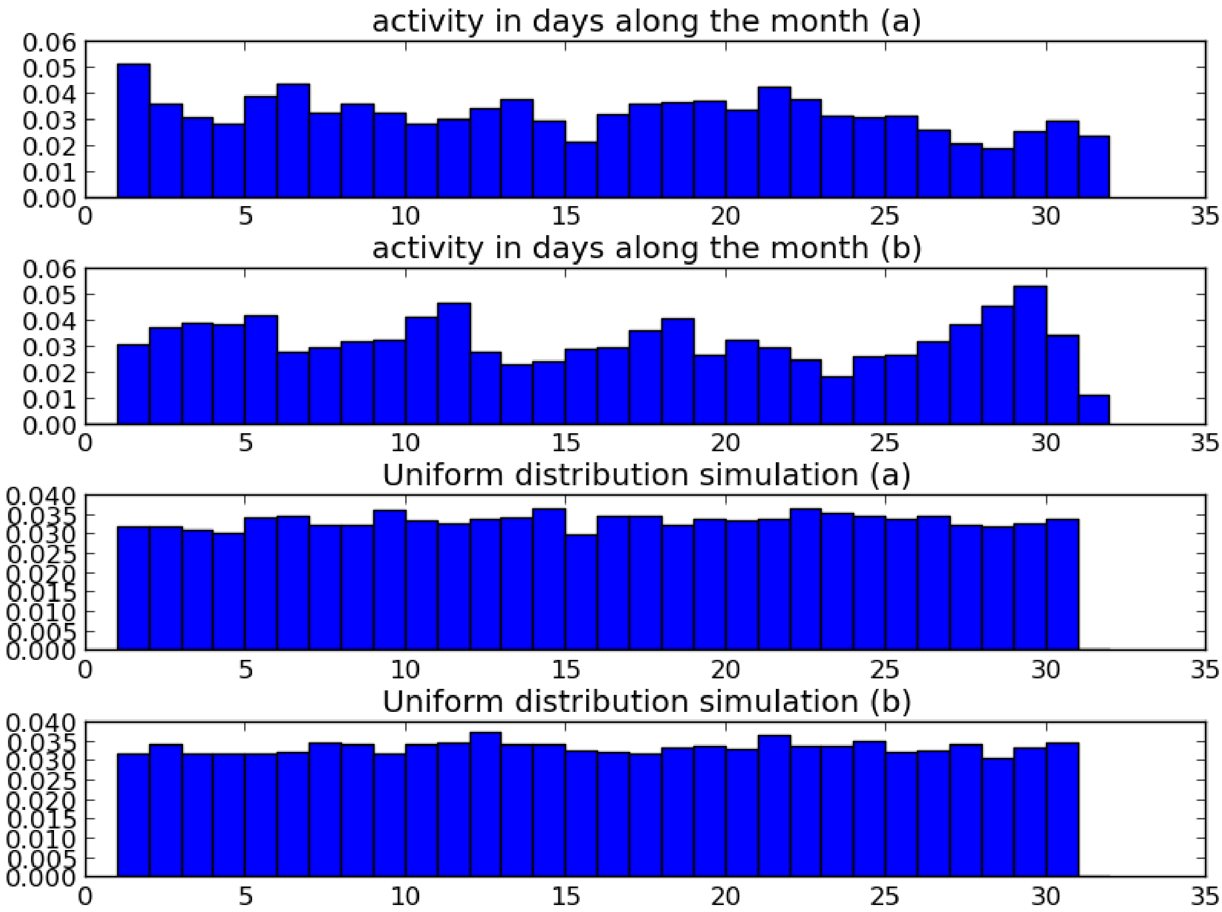

Figure 6: Histogram of AA activity along days of the month. Discrepancies in extreme incidences are more pronounced than observed with simulation using uniform distributions. While simulations delivers $\tau = \frac{max[count(i)]}{min[count(i)]} \approx 2.22$(a) and ≈ 2.23 (b), shouts present $\tau = 2.86$ and $\tau = 4.75$, respectively. Set (a) consists of all 15,145 messages from July, 2011 until December, 2011. Set (b) consists of all 21,055 messages from January, 2012 until January, 2015. Context (a) seem to be shifted by 1 day if compared to context (b). Context (b) presents a 7 day periodicity until day ≈ 20, where it seems extended until day 30. There is interesting to observe that although no hypothesis for this pattern was raised by authors, both months and weeks present patterns, while weekdays have no correlation with days of the month.

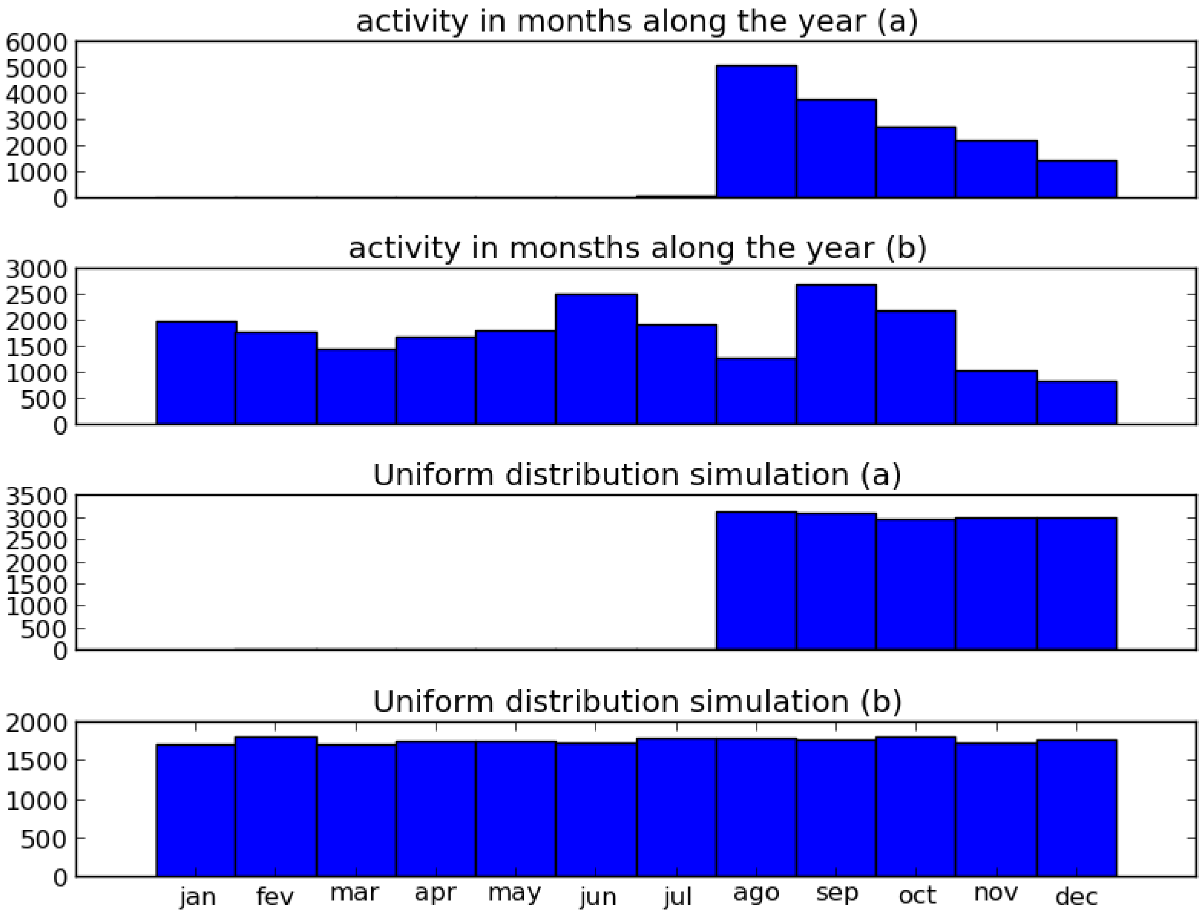

Figure 7: Histogram of AA activity along months of the year. Discrepancies in extreme incidences are more pronounced than observed with simulation using uniform distributions. While simulations delivers $\tau = \frac{max[count(i)]}{min[count(i)]} \approx 1.06$(a) and ≈ 1.09 (b), shouts present $\tau = 3.71$ and $\tau = 2.15$, respectively. Set (a) consists of all 15,145 messages from July, 2011 until December, 2011. Set (b) consists of all 21,055 messages from January, 2012 until January, 2015. Last months of the year are progressively less active. Activity seems to peak at the start of each quadrimester.

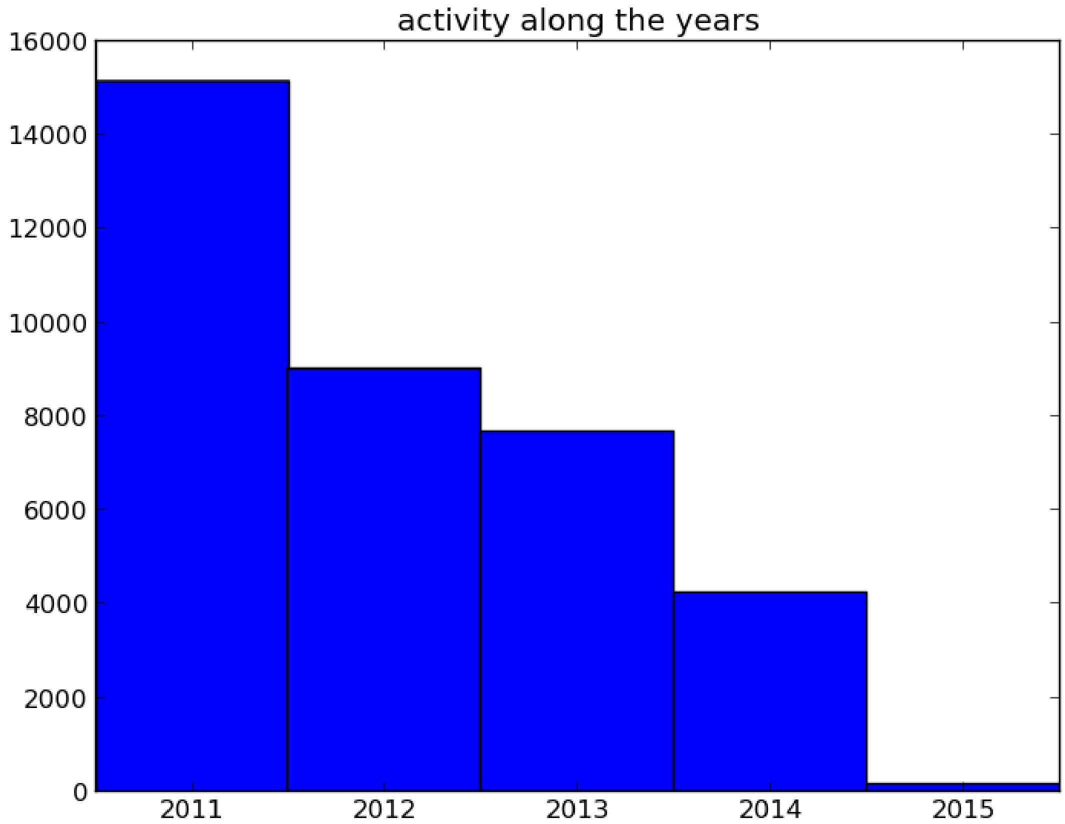

Figure 8: Histogram of AA activity along the years. Activity was most pronounced at first 6 months. Less pronounced in 2014, where there was 2 months without AA software support (see Section 4). Only first days messages are considered from 2015, as it is when the writing started.

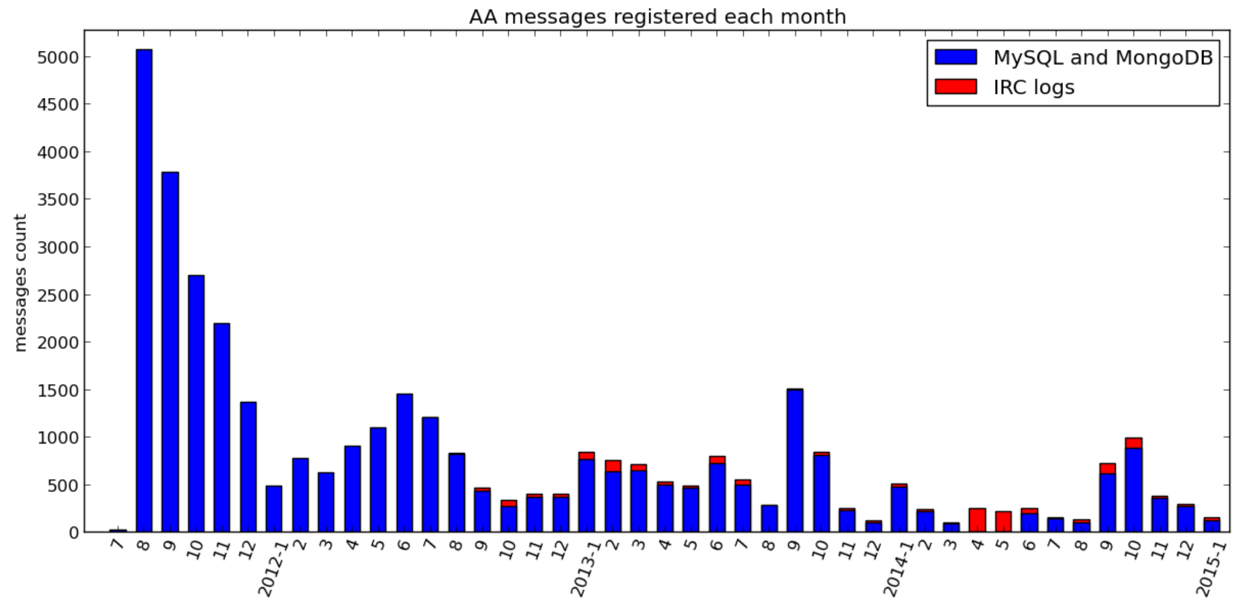

Figure 9: The average number of messages each month is $\mu = 507.1$, with a standard deviation of $\sigma = 336.63$.

6.1.2 User activity

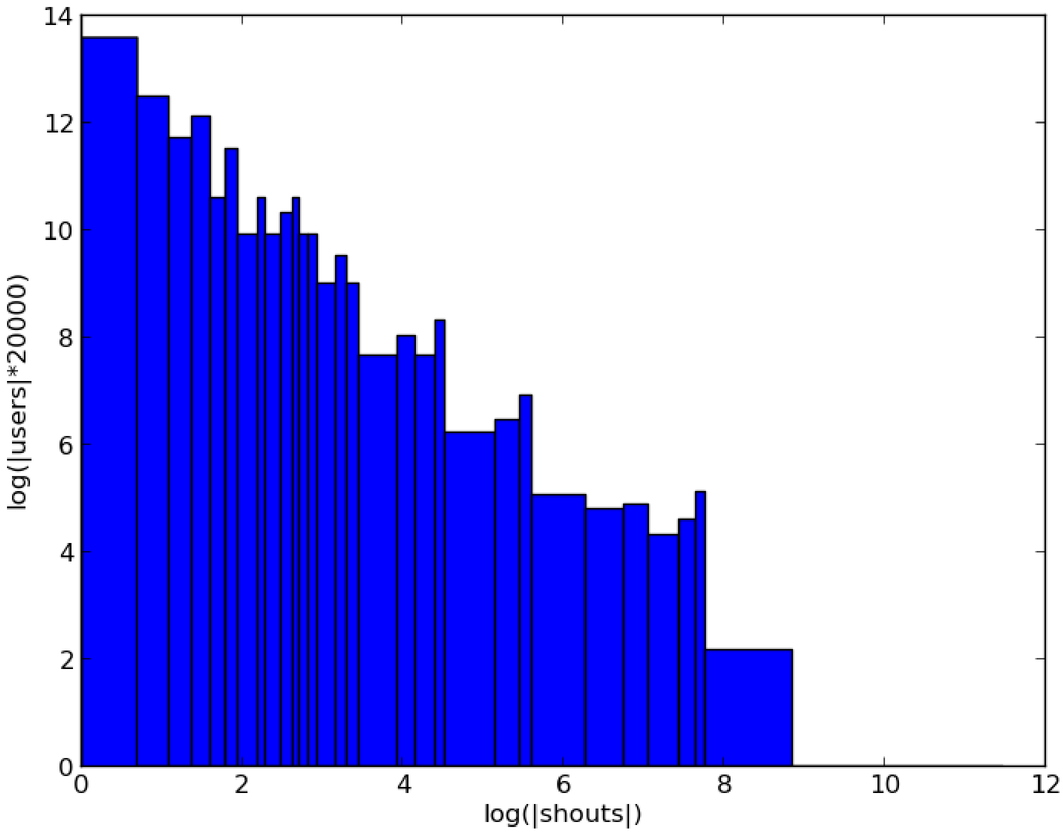

Figure 10: Histogram of user activity. The free-scale trace can be noticed by the descending line in the log x log plot.

Most active user (v1z) is responsible for 22.53% of all activity and the two most active users (v1z, o0o0o) sum up to 42.40%. The eight most active users (v1z, o0o0o, humannoise, mquasar, hick209, lari, Flecha, Fefo) sum 78.17% of activity, while 95 less active users sum 11.84%. These are in accordance with natural observations, as is the free-scale trace in Figure 10. Total number of users are 106, while (valid) messages sum 36,424.

6.1.3 Character and token incidence

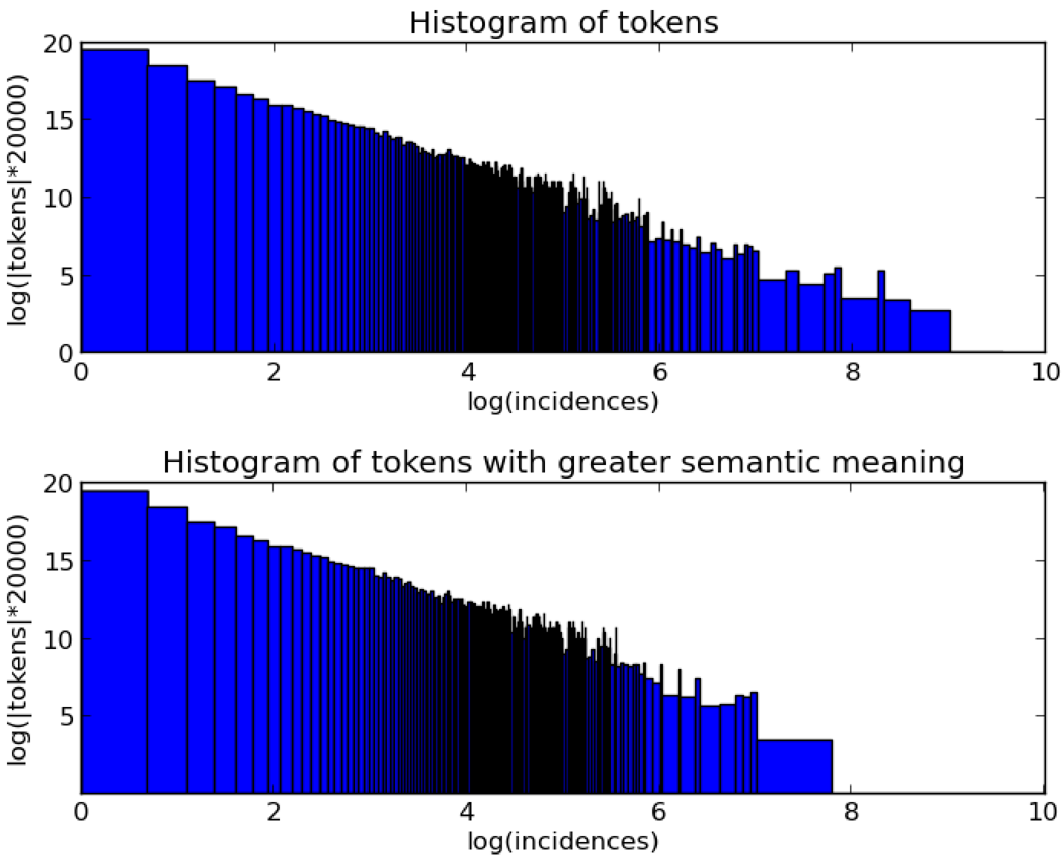

Figure 11: Histogram of token incidence in the logarithmic scale: (a) all tokens, (b) tokens which are not a English or Portuguese stopwords, s or punctuations. The Pareto trace can be noticed by the descending line in the log x log plot. Noteworthy is that theory predicts histogram (b) to exhibit a line in the logarithmic scale, but histogram (a) follows predictions more closely.

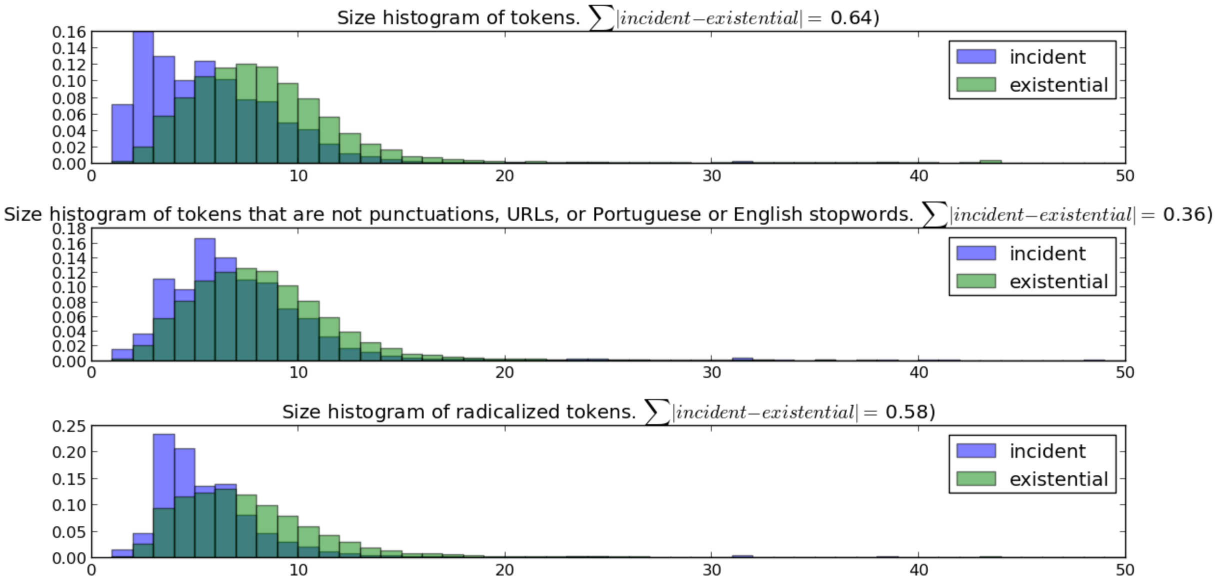

Figure 12: Histogram of incident and existent token size in number of characters. Top histogram considers all tokens; mean token size of incident tokens is $\mu_{in} = 5.74$; standard deviation $\sigma_{in} = 5.9$; mean token size of existent tokens is $\mu_{ex} =$; standard deviation $\sigma_{ex} =$. Middle histogram considers tokens which are not punctuations, s or stopwords (Portuguese and English); $\mu_{in} = 6.79$, $\sigma_{in} = 4.70$, $\mu_{ex} = 7.95$, $\sigma_{ex} = 5.00$. Bottom histogram considers all radicals from tokens used for middle histogram; $\mu_{in} = 5.57$, $\sigma_{in} = 5.02$, $\mu_{ex} = 7.89$, $\sigma_{ex} = 7.59$.

Table 4: Further observations on tokens.

description	incidence
30 most incident words	('codigo', 346), ('pouco', 347), ('AA', 355), ('paper', 357), ('dando', 379), ('wiki', 391), ('email', 413), ('escrevendo', 421), ('q', 423), ('terminando', 466), ('scilab', 497), ('p', 508), ('/Users/rfabbri/lib/notas/todos:', 511), ('eh', 541), ('aa', 596), ('ver', 613), ('indo', 620), ('lendo', 711), ('vou', 772), ('fazendo', 795), ('vendo', 906), ('fazer', 960), ('tentando', 977), ('agora', 1000), ('pra', 1058), ('nao', 1075), ('ainda', 1118), ('sobre', 1605), ('commit', 2472), ('git', 2535)
30 most incident radicals	('volt', 458), ('feit', 490), ('arrum', 506), ('p', 508), ('/users/rfabbri/lib/notas/todos:', 511), ('trabalh', 513), ('scilab', 540), ('eh', 546), ('nov', 580), ('algum', 589), ('escrev', 603), ('ver', 616), ('ind', 651), ('ach', 708), ('email', 741), ('lend', 858), ('test', 929), ('vou', 956), ('aa', 974), ('vend', 986), ('agor', 1080), ('pra', 1086), ('nao', 1114), ('aind', 1206), ('termin', 1274), ('tent', 1335), ('sobr', 1611), ('faz', 1882), ('git', 2544), ('commit', 2579)
number of web addresses [37]	1431 (1063 unique)
number punctuations	826
number Portuguese stopwords	65,090
number English stopwords	23,021
total number of tokens	255,662

6.1.4 Morphosyntactic incidence

6.2 Dependent activity

6.2.1 Time and user dependent activity

6.2.2 Language and user dependent activity

6.2.3 Language and time dependent activity

6.2.4 Time-related stability

6.3 Network activity

6.3.1 Time user networks

6.3.2 Lexical user networks

6.3.3 Network measures

6.3.4 Network primitive sectioning

6.4 Principal components formation

6.5 Immediate clustering

6.5.1 Users clustering

6.5.2 Words clustering

6.6 Timeslot clustering

6.7 Comparative analysis

6.8 AA, OCD, and Participa.br

7 Results

Although a weak evidence, the closer fit to natural predictions of using all tokens, including s and punctuations, instead of only a canonical set of words, raises the hypothesis that these elements are shares some levels of meaning with ordinary words, at least in virtual environments.

8 Conclusions

8.1 Further work

8.2 Acknowledgments

Vilson pelas versoes iniciais do aa e do plugin da lalenia. karmiac e bzum e andres pelo aa. Ricardo Fabbri pelo artigo Marcos Mendoça pelo paainel. Felipe Machado, por atribuir a responsabilidade de erigir um sistema de co-working nos idos tempos de 2009. Criadores do supy, django python, numpy e outras ferramentas de mineracao

References

1. Wikipedia, "Transparency (behavior) — wikipedia, the free encyclopedia." http://en.wikipedia.org/w/index.php?title=Transparency_(behavior), 2015. [Online; accessed 13-January-2015].
2. C. McBride, "Self-transparency and the possibility of deliberative politics," *Journal of Political Ideologies*, vol. 8, no. 3, pp. 289–310, 2003.
3. R. Fabbri, R. Fabbri, V. Vieira, A. Negrao, L. Zambianchi, M. Mendonça, D. Penalva, and D. Shiga, "The algorithmic autoregulation (distributed) software development methodology," *Revista Eletrônica de Sistemas de Informação*, 2013. http://sourceforge.net/p/labmacambira/paaper/ci/master/tree/editoracao-final/1702-corrigido.doc?format=raw.
4. S. Kim and J. Lee, "E-participation, transparency, and trust in local government," *Public Administration Review*, vol. 72, no. 6, pp. 819–828, 2012.
5. J. Habermas, "Religion in the public sphere," *European journal of philosophy*, vol. 14, no. 1, pp. 1–25, 2006.
6. H. H. Jensen and D. J. Hayes, "Private sector approaches to secure traceability, transparency and quality assurance in food chains," in *Int. Agric. Trade Research Consortium Symp., Bonn, Germany*, 2006.
7. A. Das Sarma, A. Das Sarma, S. Gollapudi, and R. Panigrahy, "Ranking mechanisms in twitter-like forums," in *Proceedings of the third ACM international conference on Web search and data mining*, pp. 21–30, ACM, 2010.
8. A. Das Sarma, A. Das Sarma, S. Gollapudi, and R. Panigrahy, "Ranking mechanisms in twitter-like forums," in *Proceedings of the third ACM*

international conference on Web search and data mining, pp. 21–30, ACM, 2010.

9. J. Perkins, *Python 3 Text Processing with NLTK 3 Cookbook*. Packt Publishing Ltd, 2014.

10. L. d. F. Costa, O. N. Oliveira Jr, G. Travieso, F. A. Rodrigues, P. R. Villas Boas, L. Antiqueira, M. P. Viana, and L. E. Correa Rocha, "Analyzing and modeling real-world phenomena with complex networks: a survey of applications," *Advances in Physics*, vol. 60, no. 3, pp. 329–412, 2011.

11. Wikipedia, "Cleodon silva — wikipedia, the free encyclopedia," 2015. [Online; accessed 13-January-2015].

12. labMacambira.sf.net, "AA wiki." http://wiki.nosdigitais.teia.org.br/AA#Tags_Propostas, 2012. [Online; accessed 21-January-2015].

13. labMacambira.sf.net, "Readme file of current AA." https://github.com/ttm/aa01/blob/master/README.md, 2014. [Online; accessed 21-January-2015].

14. labMacambira.sf.net, "Core code repository of AA01." https://github.com/ttm/aa01, 2014.

15. labMacambira.sf.net, "AA plug-in for lalenia/supybot (git repository)." http://sourceforge.net/p/labmacambira/lalenia2/ci/master/tree/plugins/AAbot/, 2011-4.

16. F. software community, "Supybot." http://sourceforge.net/projects/supybot/, 2000-2015.

17. labMacambira.sf.net, "Ubiquitous AA git repository." http://sourceforge.net/p/labmacambira/aa/ci/master/tree/omniscientAA/, 2012.

18. R. F. M. Begalli and G. Soares, "After all, who is yupana kernel." http://mutgamb.org/blog/Afinal-quem-e-Yupana-Kernel, 2011.

19. labMacambira.sf.net, "IRC dialog: "what is AA'."" http://sourceforge.net/p/labmacambira/aa/ci/master/tree/aa-what-is-aa, 2012.

20. R. Fabbri, "Nuvens cognitivas e a unificação da espécie humana." http://wiki.nosdigitais.teia.org.br/Cyberiun and http://cyberiun.tumblr.com/post/64607669758/nuvens-cognitivas, October 2013.

21. labMacambira.sf.net, "Client git repository of first AA'" http://sourceforge.net/p/labmacambira/aa/ci/master/tree/, 2012.

22. labMacambira.sf.net, "Server git repository of first AA'" https://gitorious.org/macambira_aa/macambira_aa, 2012.

23. labMacambira.sf.net, "Client of AA01." http://aaserver.herokuapp.com/minimumClient/, 2014.

24. labMacambira.sf.net, "Server of AA01." http://aaserver.herokuapp.com/shout?nick=oNickOuAnonimo&shout=a, 2014.

25. labMacambira.sf.net, "Lalenia bot git repository." http://sourceforge.net/p/labmacambira/lalenia2/ci/master/tree/, 2011.

26. labMacambira.sf.net, "#labmacambira@freenode IRC channel." http://webchat.freenode.net/?channels=#labmacambira, 2011-2015.

27. R. Fabbri, "Ontologia de participação social," http://tinyurl.com/p2doueu.

28. *Produto 5 da consultoria PNUD/ONU de Renato Fabbri*. https://github.com/ttm/pnud4/blob/master/latex/produto.pdf?raw=true.

29. "Brazilian federal social participation portal." http://participa.br/. Accessed: 2015-01-26.

30. "Cidade democrática participation portal." http://cidadedemocratica.org.br/. Accessed: 2015-01-26.

31. R. Fabbri, "Routine to translate AA MySQL database to RDFtriples.." https://github.com/ttm/aa01/blob/master/rdf/triplificaAA.py, 2014.

32. R. Fabbri, "Routine to translate AA MongoDB database to RDFtriples.." https://github.com/ttm/aa01/blob/master/rdf/triplificaAAmongo.py, 2014.

33. R. Fabbri, "Routine to translate AA shouts in IRC log to RDFtriples.." https://github.com/ttm/aa01/blob/master/rdf/triplificaAAirc.py, 2014.

34. C. Bizer, A. Jentzsch, and R. Cyganiak, "State of the lod cloud," *Version 0.3 (September 2011)*, 2011.

35. T. Heath and C. Bizer, "Linked data: Evolving the web into a global data space," *Synthesis lectures on the semantic web: theory and technology*, vol. 1, no. 1, pp. 1–136, 2011.

36. R. F. et al., "Stability in human interaction networks: primitive typology of vertex, prominance of measures and activity statistics," *arXiv*, May 2014. http://arxiv.org/abs/1310.7769.

37. LabMacambira.sf.net, "1431 links found in AA shouts.." https://github.com/ttm/ensaaio/blob/master/data/linksAA.txt, 2015.