UNIVERSITÀ
DI TRENTO

Dipartimento di Ingegneria e Scienza dell'Informazione

Corso di Laurea in
Informatica

Elaborato finale

Titolo

Sottotitolo (alcune volte lungo - opzionale)

Supervisore
.....

Laureando
Gandelli Alessio

Anno accademico .../...

Ringraziamenti

...thanks to...

Contents

Sommario

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.

Sommario è un breve riassunto del lavoro svolto dove si descrive l'obiettivo, l'oggetto della tesi, le metodologie e le tecniche usate, i dati elaborati e la spiegazione delle conclusioni alle quali siete arrivati.

Il sommario dell'elaborato consiste al massimo di 3 pagine e deve contenere le seguenti informazioni:

- contesto e motivazioni
- breve riassunto del problema affrontato
- tecniche utilizzate e/o sviluppate
- risultati raggiunti, sottolineando il contributo personale del laureando/a

1 Introduction

Wikipedia is the biggest source of information currently available on the internet, there are more than 6 million articles and they are all maintained by volunteers. The value of Wikipedia is all in the hands of the editors.

Millions of users edit Wikipedia everyday. Many users mean many different ideas and therefore the probability af a disagreement is very high so conflicts are on the agenda. Avoiding these conflicts is the best way for this encyclopedia to grow.

Each Wikipedia page has four different tabs:

- Article: the actual content of the page.
- Talk Page: a forum where people can talk about edits.
- History: a place where everyone can see the older versions of the pages.
- Source: in this section users can edit the page.

Conflicts could happen both on the Talk page, through a discussion, and in the Article, through an edit war. It is valuable to analyze all of these aspects to get a well-rounded view of the problem.

1.1 Main Project

The project our team is working on, in collaboration with Eurecat and the Wikimedia Foundation, is named: "Community Health Metrics: Understanding Editor Drop-off". this is an excerpt of the project idea:

"The primary value of Wikipedia is the editors. When an editor leaves the project, we lose their participation and contribution to the community, This could be related to multiple factors, also external to the project, but it could signal an issue related to internal dynamics and to the health of the community. While a big effort was dedicated to retain new editors, we lack knowledge and initiatives focused on understanding and preventing drop-off for experienced editors."
As stated in the project description, the focus is on expert users, who are the core of Wikipedia: there are 41,741,926 Wikipedia accounts but active users are only 132,916, namely
3% of all users.

Focusing on this category of users and understanding the reasons that lead to a drop-off can give a big help to Wikipedia. Several people are working on this project, this work is just a part of the whole. In the team, everyone is working on a specific topic with the idea of then merging the different results to obtain an analysis of the phenomenon from different points of view in order to have a greater understanding of the life cycle of users.

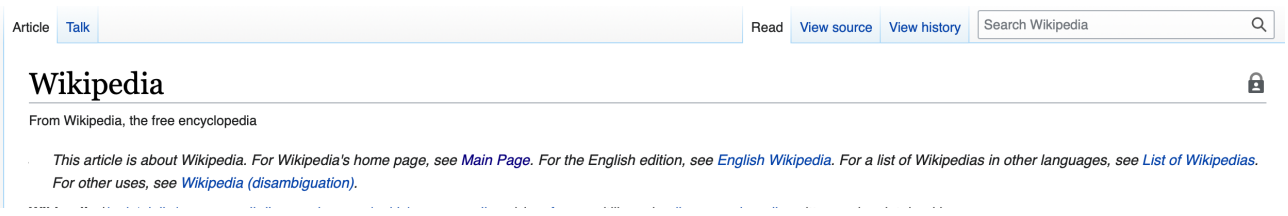


Figure 1.1: Page structure.

The prevention of the drop-off is not the only goal of the project, improving the community health is also important to let users be in a good environment without being held back from editing.

1.2 My Contribution

The topic explored in this study is the revert analysis - i.e., when the version of a page is restored to that of a specific date - for all the articles of Wikipedia.

This project consisted of the analysis of the edit history of different language editions of Wikipedia to study patterns of reverts and edit wars to understand their potential effect on individual user activity.

We implemented state-of-the-art metrics of controversy based on reverts and mutual reverts and developed a new metric based on revert chains. Metrics have been computed per page and per user monthly.

1.3 Related Work

There are several works involving reverts: An interesting tool that allows visualizing conflicts is the one developed by Suh *et al.* [2]. The problem is that it is from 2007 but Wikipedia started to grow a lot since then; now we have new technologies and much more data to analyze so more interesting conclusions can be reached. There have been analyses of antisocial behavior caused by vandalism [1], but since the focus of the project is on experienced users, this is not relevant to this study. In 2014 Yasseri *et al.*[3] had defined a controversy metric M based on mutual reverts, in this study we used this metric as startng point to develop another controversy metric G based on reverts chain. We computed metrics for both page and users including M.

2 Background

Everyone knows what is Wikipedia and how to read an article, but there are many features that most people are not aware of, *e.g.* see all the versions of a page and being able to edit it. Anyone with a browser and without much effort can see and compare all the edits in a Wikipedia page. For developers, there are many powerful resources such as big datasets containing a lot more information.

2.1 History Exploration

In the history section of a wikipedia article is possibile to see every version of the page. There are several tools anyone can use to explore the revision history:

- Mobile application: this resource is only available on mobile device and provides us some statistics about the edits of the page like the total number of revisions (Fig 2.1).

- Website: it is possible to compare two versions with an interactive tool that shows the progress of the modified page: each change corresponds to a bar indicating the number of bytes added or removed from the revision (Fig. 2.2).

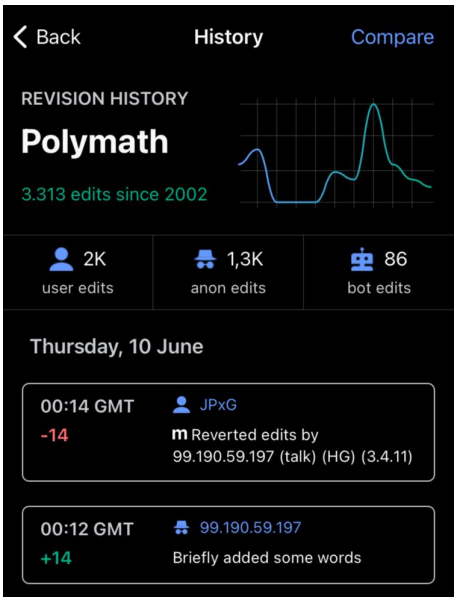


Figure 2.1: Mobile interactive visualization of the history.

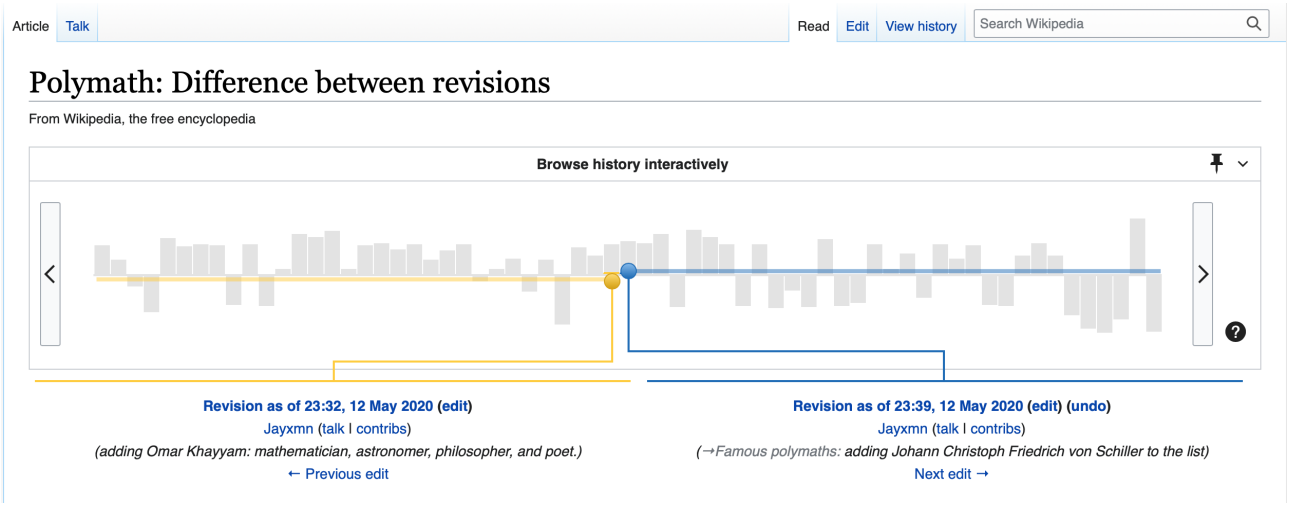


Figure 2.2: Interactive visualization of the history.

2.2 Dataset

There are two datasets that store information about Wikipedia edits made available from the Wiki-Media Foundation: a) the MediaWiki History[1] and b) the MediaWiki History Dumps [2]

The only difference other than the format (XML the former, TSV the latter) is that the former has the page content. The dataset used in this study is the MediaWiki History Dumps.

Each line of the TSV represent an event and, since it is denormalized, the events for user, page and revision are stored in the same schema. All event entity have different event types:

- Page: create, delete, move, reatore, merge
- User: create, rename, altergroup (change user rights), alterblocks (block user)
- Revision: create (edit a page)

In this analysis only revision events are of interest, there are 68 fields but only a few were needed. The entry could be divided in different sections: one section with general information of the revision like timestamp and comment, a section with information about the user who did the revision, one for the page where the revision was made, and the last one with more specific information about the revision. The most relevant fields of each section are represented in the Tables 2.1, 2.2, 2.3.The caption include, if needed, the descriptions of the fields.

id	username	groups	is_anonymous	registration	revision_count
42081	Checco	autopatrolled	False	2006-02-10 14:52:44.0	10420

Table 2.1: Data about the user who did the revision. The *groups* field helps to identify if the user is an admin. The *revision_count* is needed to calculate complex metrics like M and G.

id	title	namespace	revision_count
116530	Pino_Rauti	0	195

Table 2.2: Data about about the page where the revision took place. The *namespace* field is used to filter only the revisions from the namespace 0, i.e., the actual encyclopedia.

[1] https://wikitech.wikimedia.org/wiki/Analytics/Data_Lake/Edits/MediaWiki_history
[2] https://wikitech.wikimedia.org/wiki/Analytics/Data_Lake/Edits/Mediawiki_history_dumps

id	parent_id	is_reverted	reverter_id	is_reverter
73507165	73506955	True	73511400	False

Table 2.3: Data about the revision itself, we are able to identify if the revision is reverting another one, if it is been reverted and who is the reverter.

language	size
English	540 GB
Spanish	72 GB
Italian	54 GB
Catalan	12 GB

Table 2.4: Size of the dataset in different languages.

2.3 Definitions

It is worth defining some terms that will be used several times in the discussion.

Definition 1 *(Revert) On Wikipedia, reverting means undoing or otherwise negating the effects of one or more edits, which results in the page (or a part of it) being restored to a previous version.*

Definition 2 *(Revert chain) On a Wikipedia page, a revert chain occurs when an edit that reverts an edit is itself reverted.*

Definition 3 *(Mutual revert) A "mutual revert" is recognized if a pair of editors (x, y) is observed once with x and once with y as the reverter [3].*

Definition 4 *(Editor weight) The weight of an editor x is defined as the number of edits N performed by him or her [3].*

Definition 5 *(Mutual revert weight) The weight of a mutually reverting pair MW is defined as the minimum of the weights of the two editors [3].*

Definition 6 *(Chain weight) The weight of a revert chain CW is defined as the minimum of the weights of the editors involved in the chain.*

2.4 Metrics

Two complex controversiality metrics have been computed in this study: the first one, M, is a state of the art metric introduced by Yasseri *et al.* [3] which give us a score of the controversiality of the page based on the presence mutual reverts. The second one that we designed, called G, is very similar to M, but instead of using mutual reverts, it uses revert chains to evaluate the controversiality of the page.

Controversiality M The controversiality M of an article is defined by summing the weights of all mutually reverting editor pairs, excluding the topmost pair, and multiplying this number by the total number of editors E involved in the article.

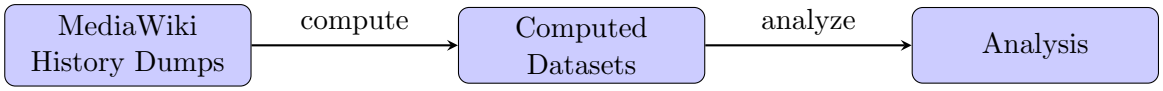
$$M = E \sum_{all\ mutual\ reverts} MW \tag{2.1}$$

Controversiality G The controversiality G of an article is defined by summing the weights of all the chains there are on a page and multiplying by the total number of editors N involved in at least one chain.

$$G = N \sum_{all\ revert\ chains} CW \tag{2.2}$$

3 Methods

Considering the huge size of the dataset and the fact that a large portion of its content was useless, smaller datasets have been computed with the aim of expediting the analysis even for future usages. The analysis was made based on the computed datasets. These datasets can be computed for every language thanks to a bash script, in this way a multilingual analysis on the most controversial topics can be conducted in different locations.



3.1 Computed Dataset

After the first skimming, only the revisions involving a revert were saved. This dataset, whose schema is the same as the MediaWiki History Dumps, has been sorted by both page and timestamp, and thanks to this screening, the size is now $\sim 10\%$ of the original. In order to achieve this result, the compressed dataset has been decompressed line by line on the fly and only the entries we were interested in have been saved in a file. Therefore only a small amount of RAM and disk space is required since all data is compressed. For the sorting part the most optimized way to sort a file, which is Unix sort, was used.

From this filtered dataset several smaller datasets have been computed, and these can be divided into two modules:

- Chains: the focus was on detecting revert chains in the pages
- Group: the focus was posed on the number of reverts that users made or received based on the groups they belong to (admin, registered, anonymous).

3.1.1 Chains

The data concerning revert chains have been computed from the compressed filtered dataset. Every time the filtered dataset was analyzed, it was read line by line and only the interesting pieces of information were saved. The output is a JSON file, in which every page corresponds to a JSON object. A list of chains and some statistics have been saved for each page. Every chain has a start and an end date, a list of revisions, and the name of the involved users. The resulting dataset is way smaller than the initial one so it is possible to browse it in only a few seconds.

In order to identify a chain, we used a function, called *simple_chains*, that differs from another one, called *complex_chains* because it identifies a chain of revert only considering contiguous reverts. We decided to use the simple one because we were only interested in those chains that occur in a short time span, since there is where most of the discussions take place. If more than 50% of users involved in a chain were bots the chain was excluded. There are two versions of this dataset, one of which considers anonymous users and one that does not.

In the schema below there are all the fields in a page object.

```
{
    "title": "Loligo_vulgaris",
    "chains":
    [{
        "revisions": ["113715375", "113715381", "113715393"],
```

```
            "users": {"62.18.117.244": "", "Leo0428": "17181"},
            "len": 3,
            "start": "2020-06-15 22:16:23.0",
            "end": "2020-06-15 22:17:38.0"
        }],
        "n_chains": 1,
        "n_reverts_in_chains": 3,
        "n_reverts": 38
        "mean": 3.0,
        "longest": 3,
        "G": 0,
        "M": 0,
        "lengths": {"3": 1}
    }
```


The user object is very similar, but it is calculated with another procedure. All the data we needed was stored in the JSON pages. By analyzing that file all the chains in which a user has been involved can be extracted, and then statistics can be calculated in a similar way as for pages. Using this dataset it can be computed 10 times faster.

The only difference is that the M field is missing because it is only related to a page, while the G field can be computed on a user considering every chain in which it is the author of at least one revision.

The dataset was also computed monthly for both users and pages, the schema is simpler than the JSON one and this allows us to save it in a TSV using only one row for each month. Instead of saving all the data regarding the chain, only the numbers of chains longer more than 5, 7, 9 were saved. In Table 3.1 there is a sample page entry. In order to do this, the JSON dataset has been processed one page (or user) at a time, after it was divided by month. The chains were counted per month basing on the start date of the chain.

title	**year_month**	**n of chain**	**n rev in chain**	**mean**	**longest**	$\geq$ **5**	$\geq$ **7**	$\geq$ **9**	**G**
Franz_Kafka	2018-11	11	113	10.3	51	4	4	3	0

Table 3.1: Entry of the mothly TSV

3.1.2 Group

Another interesting part of this study was focusing on the category a user belongs. Thanks to this we were able to track the habits of the users, and this can allow us to understand, for example, if someone stopped editing Wikipedia after several reverts from admins. Detecting these kinds of patterns is useful for community health. The groups to which users can belong are:

- Admin (sysop): can perform certain actions like blocking users and editing protected pages,
- Registered: are logged in at the time of the edit,
- Anonymous: are not logged in and their username is their IP address(it is not possible to match an IP with a user because the IP can change over time).

The datasets computed are both for pages and users:

Pages For each page, there are two topics of investigation: reverts and mutual reverts. An entry of the dataset is a page-month containing the number of reverts and mutual reverts made on the page divided by group. This can be helpful, for example, to detect pages where admins are more active and this could be a sign that something is wrong with the page.

The notation *adm_reg* in Table 3.2 refers to the number of admin that performed a revert to a registered user (similarly with *adm_adm, reg_adm, reg_reg*).

The notation *mut_ra* in the Table 3.3 refers to the number of mutual reverts where the pair is composed by a registered user and an admin. The order of the user does not matter, in fact, there is no *mut_ar* that would have the same value.

Since the focus was on experienced users, only pairs involving registered and admins were computed. For having an idea of the volume of the reverts made by anonymous we saved the number of reverts that were made by both anonymous (*anon*) and not anonymous (*not_anon*).

To compute these metrics simple variables have been used. They have been incremented, if necessary, at each entry of the dataset and they have been initialized each time a new page started. For both users and pages, we have discarded edits that have been marked as vandalism and edits made by bots.

id	**page**	**year_month**	**adm_adm**	**adm_reg**	**reg_adm**	**reg_reg**	**anon**	**not_anon**
1	AS_Roma	2020-10	14	245	36	308	1493	603

Table 3.2: Entry of the revert page TSV.

Mutual reverts are not as easy to compute as reverts. We need to store information of the whole page in order to correctly detect all the mutual reverts.

The most efficient way to save such information is using dictionaries. For each reverter has been saved the list of users who reverted. At the time of processing the page the saved information allowed us to compute mutual revert pairs.

id	**page**	**year_month**	**M**	**mut_aa**	**mut_ra**	**mut_rr**	**anon**	**not_anon**
1	Giorgio_Napolitano	2020-07	7681159	0	4	3	61	7

Table 3.3: Entry of the mutual revert page TSV.

User It is useful also to have the data aggregated by user. Reverts data can be retrieved from the filtered dataset sorted by timestamp. The data about reverts is gathered and processed month by month. We store, for each user-month, the number of reverts made and received divided by group.

When a user performs a revert, thanks to the Wikimedia History Dumps, we can know the id of the revision which is reverting but not the id of the reverted user. To solve this problem we had to save the info in different dictionaries: *reverters, editor, groups*,

reverters[username] gives us the list of the revision it reverted.
editor[revision_id] gives us the user who performs that edit.
groups[username] gives us the groups a user belongs.

Combining this dictionaries we have all the data necessary to compute all the metrics we need.

user	**group**	**year_month**
carlos	adm	2020-10

received	**r_reg**	**r_not**	**r_adm**	**done**	**d_reg**	**d_not**	**d_adm**
13	12	42	0	13	12	42	0

Table 3.4: Entry of the mutual user TSV.

The mutual revert analysis was harder to implement because in order to save the information about mutual reverts we need the dataset sorted by pages, but to get the data monthly we should use the one sorted by timestamp. We solved this problem by storing the user-page-month in the dataset, i.e., the information about the mutual reverts of a user in a specific month on a specific page. This led to a larger dataset but with a higher level of information: it is easy to post-process it grouping by user or month to have one entry per user or one entry per month, respectively.

user	**group**	**page_name**	**year_month**	**mut_adm**	**mut_reg**	**mut_not**
khalu	adm	Barcelona	2020-10	13	12	4

Table 3.5: Entry of the mutual user TSV.

4 Results and Discussion

The second step of this work was the analysis of the generated datasets. Thanks to the structure and the heavy pruning the analysis of these datasets was fast, this allowed us to have a better workflow without any interruption. We analyzed the data in two ways: a descriptive statistic and an interactive one.

Descriptive For each dataset, there is a script that plots various statistics using the python libraries Pandas and Matplotlib. There are two types of output: plots and rankings. Plots are useful to understand the trend from a more comprehensive point of view and on a monthly base. Rankings are instead used to see the pages/users ordered in a more specific way by one of the metrics previously computed.

Interactive We decided to make an interactive dashboard available online. The idea is that everyone can change a few parameters and see how the metrics are performing in a personalized way. To achieve this we uploaded our dataset on a database and thanks to an innovative way to retrieve data (grapQL) we can display it on a website.

Generic statistics As we can see here the biggest part of the pages has 0 reverts. Since filtering the Wikimedia History Dumps removed all the pages with 0 reverts, this field has been computed by subtracting, from the total number of pages, a value that is available on Wikipedia [1].

n_reverts	**n_pages_it**	**n_pages_ca**
0	1.296.915	626,5326
1	186,539	32,233
2-4	122,072	15,387
5-9	45,391	4,791
10-99	47,833	3,906
100-999	4,145	84

Table 4.1: Number of reverts for Italian and Catalan Wikipedia.

4.1 Chains

Thanks to the analysis of the page chains we can have an overview of an entire Wikipedia in a language, discovering statistics like the mean length of chains or the longest one. Another aspect worth investigating was the relationship between solitary reverts and reverts that are in a chain: more reverts in chains mean more discussions. In cases like these combining the data of the other team members who analyzed the talk pages could be useful to better understand the dynamics. While the chains in the pages are useful to have a less specific but wider view of the phenomenon, studying the chains a user joined lets us see if a specific user is involved in many chains and in which pages is more active. In this sense, we can define different categories of users: the ones who are active just in some topic or the others who revert on all Wikipedia.

Monthly metrics are even more interesting: we can plot the trend of reverts on a page and see if it is always controversial or just in a specific historical moment related to something that happened in

[1] `https://en.wikipedia.org/wiki/Special:Statistics`

the world. Plotting the metrics year by year allows us to understand the global activity of the users on Wikipedia. We can define the lifecycle of a user and see when it is more active, and if its decrease of revisions is related to a discussion.

In Table 4.2 there is an overview of the number of solitary reverts and reverts which belong to a chain in the Wikipedia of different languages. The ratio between the number of reverts that are in a chain and the ones that are not is a useful indicator of how much the users are committed. In the Catalan Wikipedia this ratio is higher and we could refer this to the patriotism that brought more attention on certain topics that we will explore later in Table 4.3.

len	revisions	reverts	reverts on edits	reverts in chain	% in chain
en	1,027,188,756	66,147,314	6.4%	6,144,948	10%
es	136,318,137	11,539,552	8.4%	1,065,618	9%
it	121,362,136	7,712,039	6.4%	850,020	11%
ca	27,657,030	355,251	1.3%	56,280	15%

Table 4.2: Number of reverts in Wikipedia in Spanish, Italian, and Catalan.

4.1.1 Page

In Table 4.3 the pages are ranked by the number of chains in Italian, Catalan, and Spanish. In the Italian one six out of ten pages were football-related while in the catalan one we can see, as expected, a stronger territorial belonging, the Spanish ranking tells us that the main part of Spanish Wikipedia users are from Latin America and that they are interested in football. It is interesting to note how in the Catalan Wikipedia the second surname of the person which the article is about is written in the title of the article while in the Spanish one it is not.

id	title	chains	title	chains	title	chains
1	Serie A	195	Barcelona	68	Club América	222
2	Juventus FC	190	FC Barcelona	33	Deporte en Argentina	218
3	Matteo Renzi	179	Catalunya	30	Club Universitario	213
4	AS Roma	176	País Valencià	26	Club Guadalajara	211
5	Personale WWE	167	Marc Márquez i Alentà	22	América Latina	185
6	SSC Napoli	162	Mireia Belmonte i García	22	Club Alianza Lima	179
7	Inter	162	Girona	20	Idioma español	171
8	Roma	154	Rafael Nadal i Parera	19	Juventus de Turín	162
9	Tiziano Ferro	141	Oriol Junqueras i Vies	17	Ecuador	160
10	Gianluigi Buffon	137	Català	16	Bogotá	159

Table 4.3: Pages with more chains

In the analysis of the longest chain, the scenario we face is different. The top topics are not sports but cinema, music and literature for Italian. But here the most fascinating things happen on the Catalan Wikipedia: we can see how the longest chains are all related to Navarra, doing a more specific research we can see that it is all related to the language used to identify cities, this is probably vandalism from some Spanish user who wanted to suppress the Basque language. These metrics cannot be used for detect problems in the community health but can let some sociopolitical issues inside a place with linguistic minorities emerge.

id	title	longest it	title	longest ca	title
1	Pino Rauti	114	Roncal-Salazar	81	Alan Jackson
2	Carlos Tévez	66	Tractat d'Utrecht	80	A
3	Rogue One	64	Gazteluberri	76	Consejo Mundial de Boxeo
4	Rocky Marciano	64	Comarca de Sangüesa	71	Guerra anglo-española (1625
5	Poeta urbano	58	Comarca d'Aoiz	69	Guerra de la Independencia
6	Paradisi per illusi	55	Comarca de Lumbier	69	Guerra anglo-española (1585
7	Kuromajo-san ga toru!	53	Riu Gor	53	Independencia de la Repúbli
8	Matt Dillo	52	Tudela	51	Kreutzberger
9	Aletheia (album)	52	Igúzquiza	50	Dalas Review
10	Franz Kafka	51	Untziti	48	Bastille

Table 4.4: Pages sorted by longest chains.

monthly By analyzing the trend(Fig 4.1) of the page (Barcelona in Catalan) we can clearly see that even if there are a lot of chains the controversiality metric G grows mainly on one occasion, the reason is that in that chain experienced users are involved so this metric is useful to detect discussion between them.

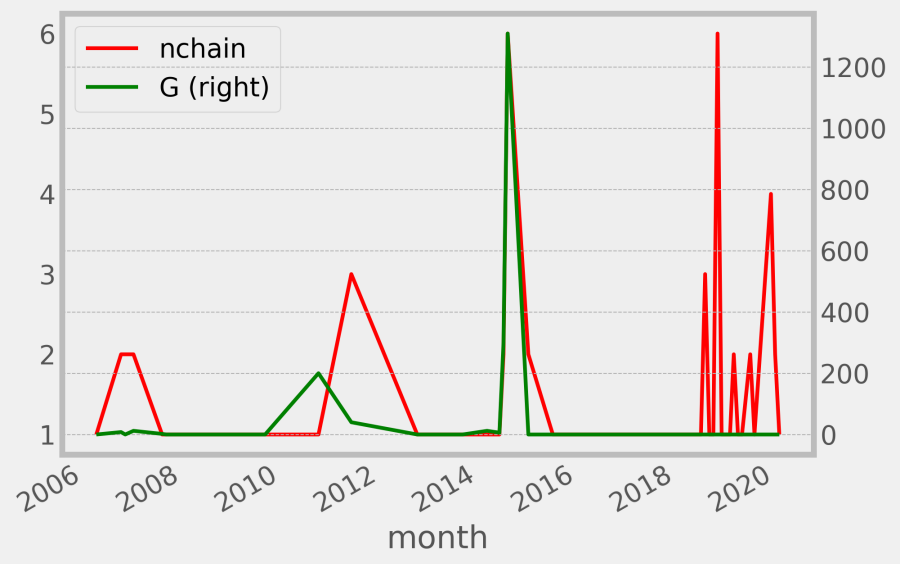


Figure 4.1: G and the number of chains for Barcelona page in Catalan.

4.1.2 User

wars Table 4.5 ranks the users by the mean length of the chains they joined. All of them are anonymous users. The longest chains are usually about details (like, for example, the number of championships won by Juventus) that are continuously modified until the vandal is blocked.

id	user	nchains	mean	nrevert
1	95.20.240.x	7	60.4	423
2	95.20.242.x	1	51.0	51
3	37.11.145.x	14	51.0	714
4	95.20.249.x	14	51.0	714
5	83.49.253.x	1	47.0	47

Table 4.5: User sorted by mean length of the chains joined

It is also possible to see on which pages a given user joins more revert wars. Table 4.6 represents the pages on which the user, let us call him Juan, got involved in more chains in The Catalan Wikipedia. In this example, Juan is mainly interested in Catalan famous people like sportsmen and writers.

id	user	nchains
1	Marc Márquez i Alentà	14
2	Barcelona	9
3	Jocs Olímpics d'estiu de 1992	7
4	Rafael Nadal i Parera	6
5	Catalunya	6
6	Lliga de Campions de la UEFA	6
7	Quim Monzó	6
8	Polseres vermelles	6
9	Jordi Sànchez i Zaragoza	6
10	Alfons Arús i Leita	6

Table 4.6: Top 10 pages by number of chain of Juan.

monthly Given a user we can draw its revert chain activity.

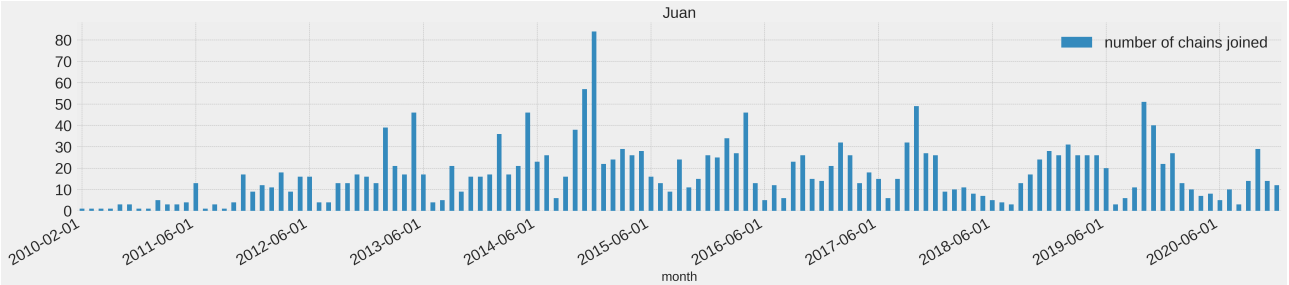


Figure 4.2: Number of chain by month of Juan

4.2 Group

From the analysis of the groups, we can define different rankings of pages using the number of reverts of each group. Given a page, we can plot the trend of the edits by group and detect the pages in which the admins are more interested. We can say from which category a given user is the target of reverts and the ratio between made and received reverts. Such a deep analysis of this data can be done, that is the reason why it is available to everyone who needs it.

Here are some numbers about the users in different languages:

len	registered	admin	active
en	41,825,139	1,089	127,566
es	6'266'812	69	16'143
it	2'140'498	114	8'208
ca	391'067	22	1'180

Table 4.7: Number of users by group.

4.2.1 Page

reverts In Fig 4.3 we can see how the influence of the admins is higher in the Italian Wikipedia than in the Spanish one. All the spikes we can see in Spanish and Catalan Wikipedias are due to the seasonality of the user activity: every year during the summer there is a decrease in the edits and therefore of the reverts. In Catalan, these trends are more visible, and we can see that the number of reverts made by admin towards registered users is similar to the ones done by registered toward registered unlike the Italian and Spanish ones.

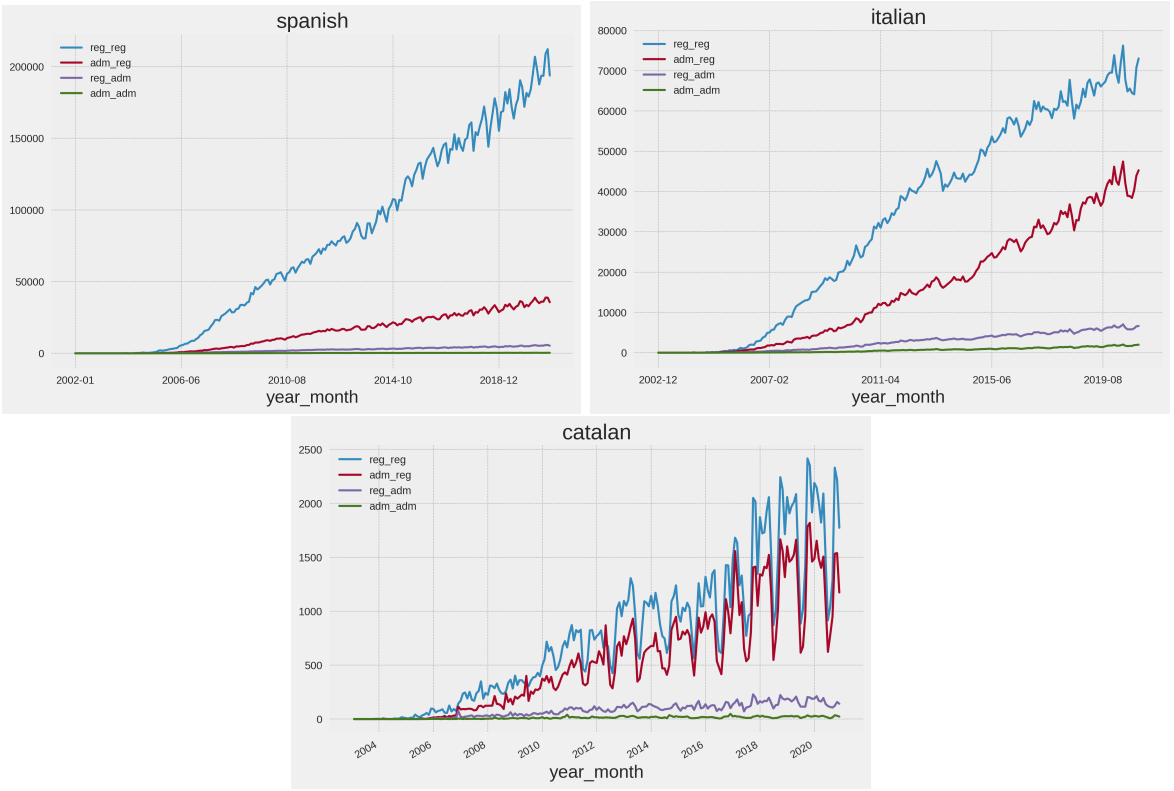


Figure 4.3: Number of reverts done divided by group.

mutual In this graph, we can see a comparison between M and the total number of reverts done and we can notice that M had a big growth until 2019 where it remained stable.

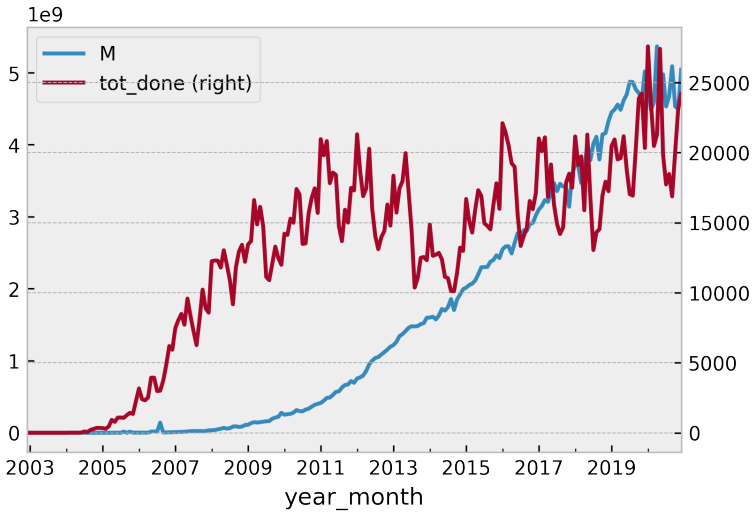


Figure 4.4: mcompare

4.2.2 User

From the analysis of the reverts devided by group we are able to see the influence of admins and the distribution of the revert done and received during the months. It is possible to know, for a given user, how many reverts he received and made to each category.

reverts The plots represent, for Italian, Catalan and Spanish, the number of reverts done and received by each category; we can see how the behaviour of the users changes with the language especially towards anonymous users. The Italian and the Spanish trends on done reverts are similar, but the share of the reverts of the admin in the received ones is very different. In Italy, the received reverts are equally divided between anonymous and registered.

Figure 4.5: number of chain by month of juan

5 Infrastructure

All the code is available on Github [1] where there is an organization called WikiCommunityHealth where each team member commits its contribution to the project. For the data processing, we used python since is the best option to handle that amount of data. The data is currently stored in the Unitn servers of the Cricca group.

Multi Language All the datasets computed are the results of several python scripts launched singularly. All the work has been done using Italian Wikipedia as an example. Automatizing the process allows us to run all the scripts in different languages without further effort. For achieving this automation we used a bash script that takes the language as parameters e.g *./generate_dataset it* takes the data from the Wikimedia History Dumps in Italian, create a folder "it" and all the subfolders needed and then it generate the dataset in the right place. the only requirement is that the dump must have already been downloaded.

5.1 Workflow

File For computing the datasets, since we have to process huge amounts of data, we decided to use a simple style of programming without using complex libraries. We used dictionaries which are one of the most efficient data structures. Even if the dataset computed were different we used always the same structure of the code and it consisted of few steps, the program reads the compressed Wikimedia History Dumps line by line and for each line :

1. Parse from the dataset the pieces of information.
2. Insert in dictionaries the information we want to save.
3. Check if the page id is different from the previous one, if this is true it means that a page is finished so we can process it and initialize all the variables for a new page.
4. If the page is not finished, we check if the month is finished and similarly to the page we process the information we gathered since we want to save an entry for each month.
5. If nor the page nor the month is finished we can check if this revision is reverting the previous one or doing the computation we need in that specific file.

5.2 Repository

To handle a big project is important to organize in the best way possible the repository to avoid confusion while browsing also the naming of the file is important.

Folder This is the structure of the folder where the code is organized, the bash folder contains all the bash scripts used. the utility one has all the python files that were used to check things, for example, some files let us extract from the dataset the data about a specific page. The main folder instead have all the files concerning the computing and analyzing of the datasets, the database one has the script for uploading the files on the database for the interactive dashboard

```
bash
main
    analyze
    create
```

[1] https://github.com/WikiCommunityHealth/wikimedia-revert

```
│
│   │
│   └── database
└── utility
```

Naming All the files in the main directory follow strict naming rules. The format is :
`type_class_aggregation_name_month_format.py`

- type = the file can be one that creates a dataset (c) or that analyze it (a)
- class = since this work can be divided into 2 different sections: Chain and Group so we use them as the main identifier to classify the files. It exist also a generic class used when the data computed was neither a chain on nor a group.
- aggregation = if the file is by user or by page
- name = the name of the metrics its compute
- month (optional)= if the file is by month
- format = In create files the format of the output file is written directly in the file name to fastly understand what data it handles (TSV or JSON)

5.3 Data

A lot of different files were created in the process, so they must be well organized to retrieve them without errors. There is a folder for each class: chains and groups and for both of them there is a folder for page and one for users

Folder structure Here the folder structure of how the data is stored :

```
ita
├── chains
│   ├── user
│   │   ├── wars.json
│   │   └── monthly.tsv
│   ├── page
│   │   ├── wars.json
│   │   └── monthly.tsv
│   └── page_reg
│       └── wars.json
└── group
    ├── user
    │   ├── mutuals.tsv
    │   ├── reverts.tsv
    │   └── all.tsv
    └── page
        ├── mutuals.tsv
        ├── reverts.tsv
        └── all.tsv
```

Bash Script The main code written is in python but for some of the task we decided that was better using a bash script, in particular to automatizing processes like downloading the Wikimedia History Dumps or the generation of the datasets.

6 Conclusions

This is still an open project, so the results described in this report are not complete. The biggest part of the time has been dedicated to the computation of the datasets, for the analysis defined in the project description we should combine the data of all project members. Nevertheless, with the generated datasets it is possible to draw some conclusions. We have seen how the language, and so the place, of the users, characterize Wikipedia. We have seen just some language but the study could be extended to all available languages without problems.

Future works will comprehend, other than the study in different languages, an interactive dashboard available online with this data and one of the other group members. this allows the users to dynamically retrieve the data and plot the results as they wish.

Wikipedia is full of vandals but fortunately, they are neutralised fastly. From the number of chains we can understand which are the topic in which people cannot reach an agreement; in Italy and south America these topics are sport, especially football, in Catalunya the most debated topics are about territorial belonging.

Bibliography

[1] Johannes Kiesel, Martin Potthast, Matthias Hagen, and Benno Stein. Spatio-temporal analysis of reverted wikipedia edits. In *ICWSM*, 2017.

[2] Bongwon Suh, Ed H. Chi, Bryan A. Pendleton, and Aniket Kittur. Us vs. them: Understanding social dynamics in wikipedia with revert graph visualizations. In *2007 IEEE Symposium on Visual Analytics Science and Technology*, pages 163–170, 2007.

[3] Yasseri T., Spoerri A., Graham M., and Kertész J. The most controversial topics in wikipedia: A multilingual and geographical analysis. *In: Fichman P., Hara N., editors, Global Wikipedia: International and cross-cultural issues in online collaboration. Scarecrow Press*, 2014.