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 in the English Wikipedia and they are all maintained by volunteers. The value of Wikipedia is all in the editors' hands.

Thousands of users edit Wikipedia everyday. Many users means many different ideas and therefore an higher probability of a disagreement; this means there are conflicts are on the agenda. Avoiding or reducig 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: a section where 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 these aspects to get a well-rounded view of the problem.

1.1 Main Project

The project our team, in collaboration with Eurecat and the Wikimedia Foundation, is working on is called: "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 only 132,916 users are active, nearly 3% of all users[1].

Focusing on this category of users and understanding the reasons that lead to a drop-off can give Wikipedia a big help. Several people are working on this project, because it's 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 better understanding of the life cycle of users.

[1] https://stats.wikimedia.org/EN/TablesWikipediansEditsGt5.htm

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 in 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 therefore more interesting conclusions can be reached. There have been analysis 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] defined a controversy metric based on mutual reverts, called M. In this study we used this metric as a starting point to develop another controversy metric based on reverts chains, called G. We computed metrics for both page and users including M.

2 Background

Everyone knows what Wikipedia is and how to read an article, but there are many features that most people are not aware of, *e.g.* the possibility to see all the versions of a page. Anyone with a browser and can see and compare all the edits in a Wikipedia page without much effort. 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 it's 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 devices 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 us 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).

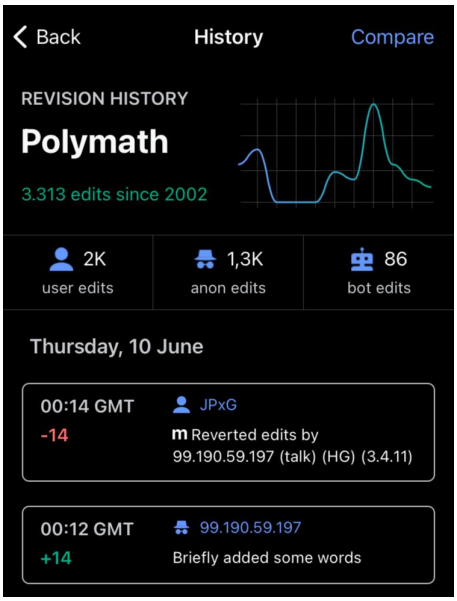


Figure 2.1: Mobile interactive visualization of the history.

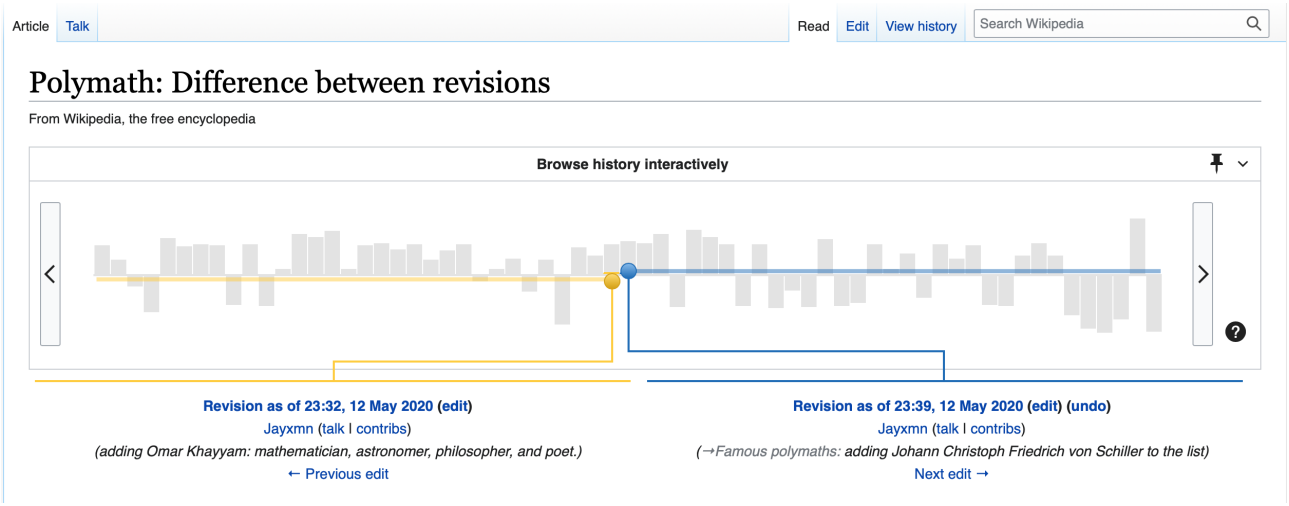


Figure 2.2: Interactive visualization of the history.

2.2 Dataset

There are two datasets that store information about Wikipedia edits, that were made available by the WikiMedia Foundation: a) the MediaWiki History[1] and b) the MediaWiki History Dumps [2]

The only difference between the two (XML the former, TSV the latter) is that the first one also contains 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. Event entities can be of different types:

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

In this analysis we are only interested inrevision events; they contain 68 fields but only a few of them are needed. The entry could be divided in different sections: one section with general information about 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 includes, 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 made the revision. The *groups* field helps to clearify 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 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 gives us a score of the controversiality of the page based on the presence of mutual reverts. On the other hand the one that we designed, called G, is very similar to M, but instead of using mutual reverts, it uses reverts chains to evaluate the controversiality of the page.

Controversiality M The controversiality M of an article is defined by summing the weights of all the mutually reverting editors pairs, excluding the topmost pair, and multiplying this number by the total number of editors E involved in the article. Using the number of edits as user weight allows us to focus on experienced users, the higher is the edit count the higher the value of the M metric will be. This is also useful to exclude the anonymous users since their edit count is equal to zero.

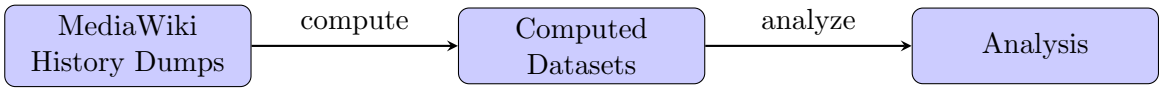
$$M = E \sum_{all\ mutual\ reverts} MW \tag{2.1}$$

Controversiality G The controversiality G of an article is defined by summing the weights of all the chains there are on a page and by multiplying this number by the total number of editors N involved in at least one chain. Similarly to M, thanks to users' edit count, the weight of chains with anonymous users involved will be 0.

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

3 Methods

Considering the huge size of the dataset and the fact that a large portion of its content is not needed, smaller datasets have been computed with the aim of expediting the analysis even for future usages. The analysis was made starting from 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 one used for MediaWiki History Dumps, has been sorted by both page and timestamp and now, thanks to this screening, the size is approximately 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 were 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 treat 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 a user made or received based on the groups it belongs to (admin, registered, anonymous).

3.1.1 Computing revert chains datasets

The data concerning revert chains have been computed starting 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 former because we were only interested in those chains that occur in a short time span, since there is where most of the edit wars 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 number of chains longer than 5, 7 and 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, then 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 Computing user group datasets

Another interesting part of this study was focusing on the category a user belongs to. Thanks to this we were able to track the habits of the users, and this allows 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 which users can belong to 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. The 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 are computed for both pages and users:

Pages For each page, there are two topics for the investigation: reverts and mutual reverts. The entries of the dataset are a page-month containing the number of reverts and mutual reverts made on the page divided by group. This can be helpful, for example, in order to detect pages where admins are more active, because 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 admins who 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. To have an idea of the fraction 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 every 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 the list of users who reverted has been saved. 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 also useful to have the data aggregated by user. Reverts data can be retrieved from the filtered dataset sorted by timestamp. The data about reverts are gathered and processed month by month. We store, for each user-month, the number of reverts both 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 information 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 to.

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 needed the dataset to be sorted by page, but to get the data month by month we should have used 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 the creation of a larger dataset but with a higher level of information: it is easy to post-process it grouping by user or month to have, respectively, one entry per user or one entry per month.

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 This section is still under contruction. We will 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.

4.1 General statistics

Number of reverts in pages As we can see in Table 4.1 in the largest part of the pages there are not any reverts. The filtering of Wikimedia History Dumps removed all the pages with zero reverts. To compute this field we needed the total number of articles in Wikipedia, a value that is avaiable on the statistics page of Wikipedia.[1]. To compute how many pages have 0 reverts we subtracted the number of pages with at least one revert from the total number of pages.

n_reverts	n_pages_it	n_pages_ca	n_pages_en	n_pages_es
0	1.296.915	626,5326	4,672,316	1,411,056
1	186,539	32,233	1,272,960	116,011
2-4	122,072	15,387	949,689	84,324
5-9	45,391	4,791	388,839	35,461
10-99	47,833	3,906	472,379	44,155
100-999	4,145	84	69,804	7113

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

M In this graph, we can see a comparison between M and the total number of reverts done and we can notice how M had a big growth until 2019 where it remained stable. The fast growth can be reconducted to the fact that M is calculated from the edit count of the users, so as the years went by the users got more experienced and then the edit count grew with a direct consequence on M.

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

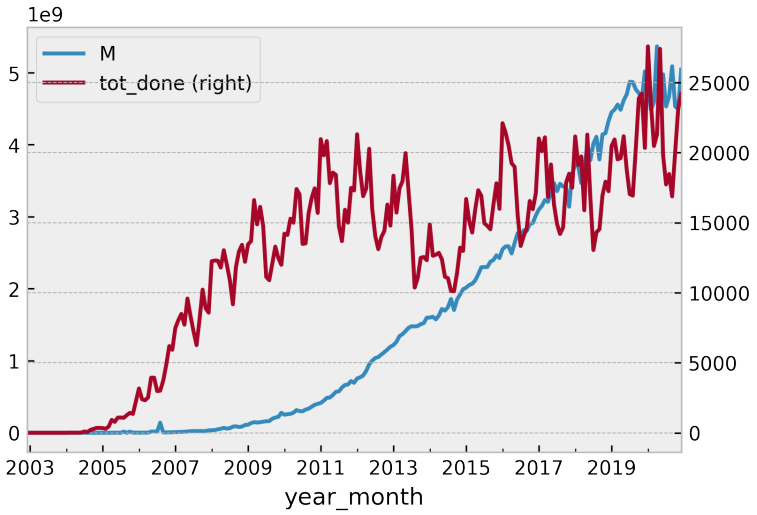


Figure 4.1: M compared to the number of Reverts. The values of the total reverts done are represented on the right Y axe.

4.2 Revert 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 the chains means more conflicts. 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 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 conflict.

In Table 4.2 there is an overview of the number of solitary reverts and reverts that 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 prone to conflicts. In the Catalan Wikipedia, this ratio is higher than in the ones in Spanish, Italian, and English; we will explore later in Table 4.3 the most conflictual topics. Further studies about the relationship between the number of reverts and the family of the language could be done.

language	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 chains in Wikipedia in Spanish, Italian, and Catalan.

4.2.1 Revert chains analysis by 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.

id	title it	chains	title ca	chains	title es	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: Top 10 pages by number of revert chains in the Italian, Catalan and Spanish Wikipedia.

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 can be used to detect problems between experienced users but can let some sociopolitical issue inside a place with linguistic minorities emerge.

id	title it	longest	title ca	longest	title es
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 (1625-1630
5	Poeta urbano	58	Comarca d'Aoiz	69	Guerra de la Independencia Espa
6	Paradisi per illusi	55	Comarca de Lumbier	69	Guerra anglo-española (1585-1604
7	Kuromajo-san ga toru!	53	Riu Gor	53	Independencia de la República Do
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 4.2) 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 the one where more experienced users are involved this metric is useful to detect conflicts between them.

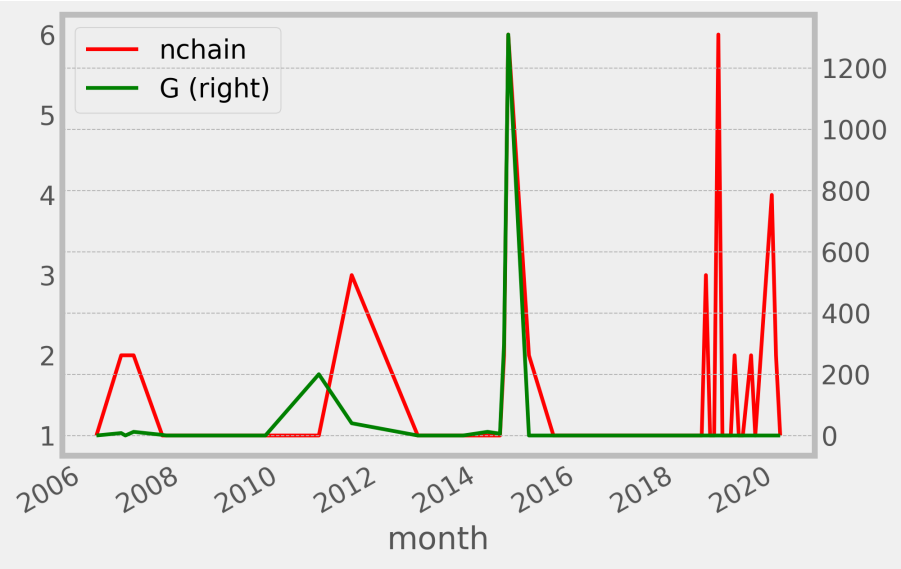


Figure 4.2: G and the number of chains for the Barcelona page in Catalan. The value of G is represented on the right Y axe.

4.2.2 Revert chains analysis by user

In Table 4.5 the data about the number of users was aggregated by the number of joined chaines, we can see that most of the users are not involed in chains. In the Italian Wikipedia more than 11% of users are involved in at least one chain versus the less than 4% of the English Wikipedia.

language	users	users in chains	ratio
en	41,840,255	1,569,240	3.7%
es	6,268,217	283,914	4.5%
it	2,141,529	248,646	11.6%
ca	391,272	12,565	3.2%

Table 4.5: Number of users in chains in Wikipedia in Spanish, Italian, and Catalan.

Table 4.6 contains the data concerning the number of users grouped by the number of chains joined.

n_chains_joined	n_users_it	n_users_ca	n_users_en	n_users_es
0	1,892,837	378,697	40.270.206	5.985.304
1	211,746	10,500	1,281,670	246,015
2-4	29,685	1,548	217,140	30,010
5-9	4,066	231	37,448	4,331
10-99	2,672	249	28,734	2,999
100-999	477	37	4’247	558

Table 4.6: Number of users that joined chains for Italian, Catalan, English, and Spanish Wikipedia.

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

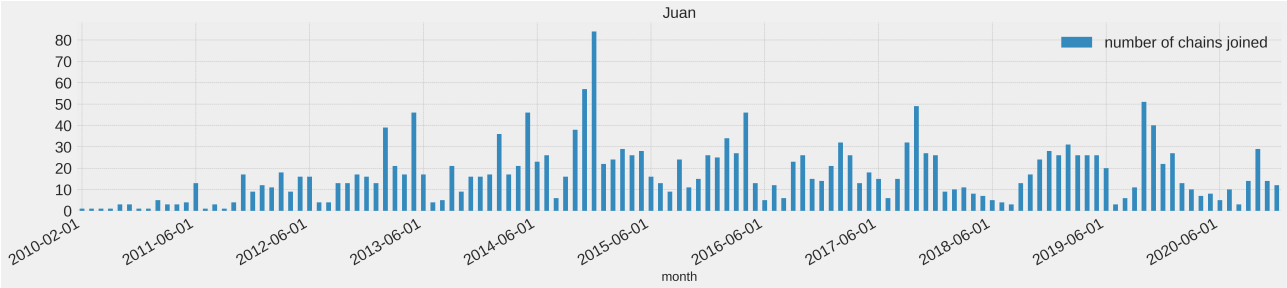


Figure 4.3: Number of chains of an anonymized user by month

4.3 User group analysis

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 active in making reverts. 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, users who have performed an action in the last 30 days are considered active:

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.

4.3.1 User activity by group

For privacy reasons we can only display aggregated data about users. This data is useful to understand the general trends a Wikipedia in a language is following. The data is divided by user category, therefore we can draw conclusions about the influence of the different groups in each language.

Mutual reverts with admins In Table 4.8 is displayed the number of users that are in a mutual revert with admin and the number of them.

mut_adm	n_users_it	n_users_ca	n_users_en	n_users_es
0	2,098,014	388,453	41,490,742	6,245,755
1	43,515	2720	342402	23104
2-4	1,300	91	7103	358
5-9	0	0	8	0

Table 4.8: Number of mutual reverts with admin for Italian, Catalan, English, and Spanish Wikipedia.

Reverts from a group to another In Fig 4.4 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 in the Italian and Spanish cases.

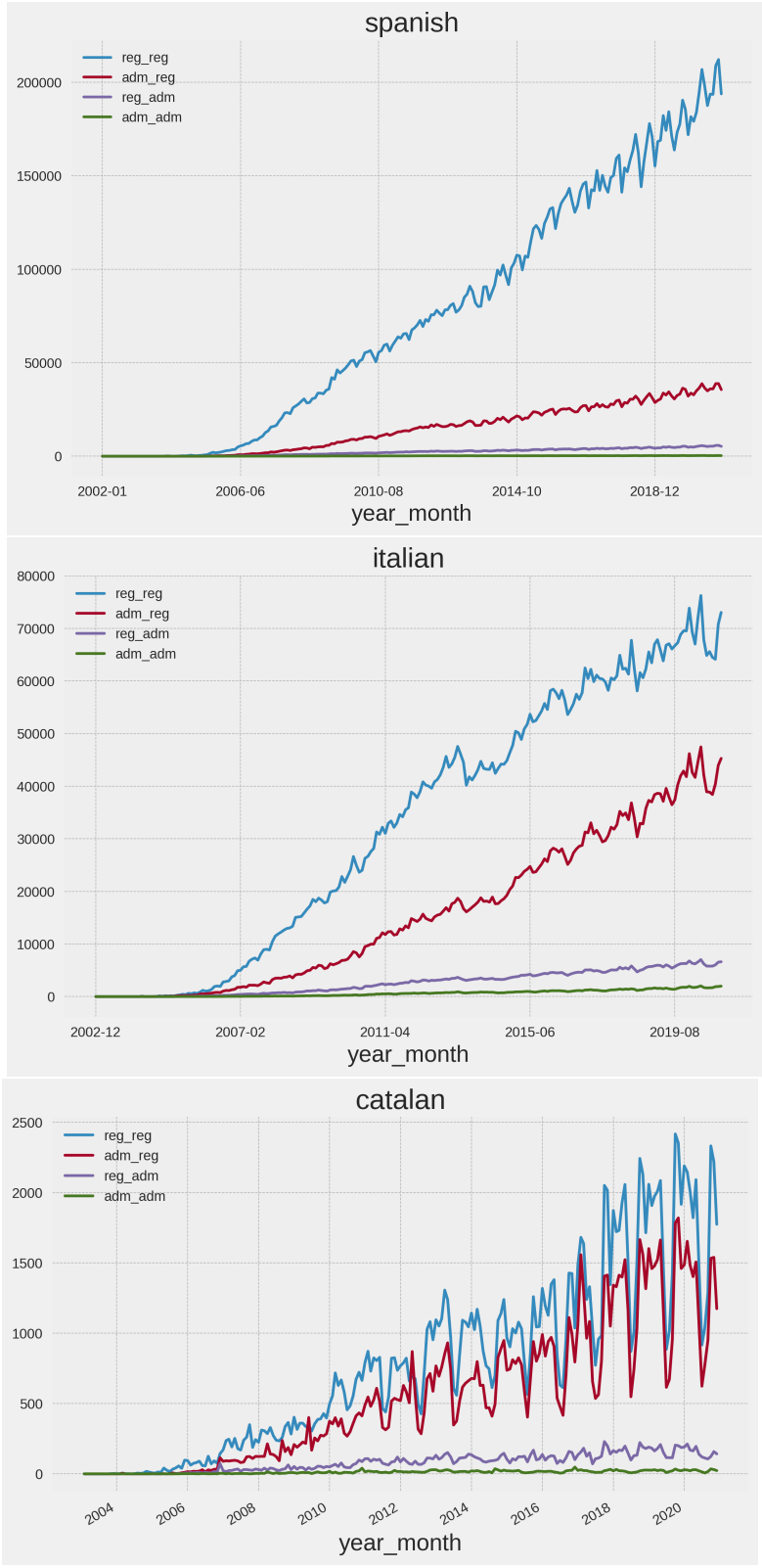


Figure 4.4: Number of reverts done divided by group.

Reverts made and received 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 reverts made and received in Spanish, Italian, and Catalan Wikipedia divided by group.

5 Infrastructure

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

Multi Language All the computed datasets are the results of several python scripts launched one by one. 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. In order to achieve 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, then creates a folder called "it" and all the required subfolders and then it generates the dataset in the right location. The only requirement is that the dump must have already been downloaded.

5.1 Workflow

File To compute 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 as an example of one of the most efficient data structures. Even if the computed dataset were different we always used the same structure of the code, that consisted in a 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 the 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 neither the page nor the month is finished we can check if this revision is reverting the previous one or doing the computation we needed in that specific file.

5.2 Repository

To handle a big project it is always 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 used bash scripts. The utility one has all the python files that were used to check various things, for example, some files let us extract from the data about a specific pagefrom the dataset. The main folder has all the files concerning the computation and analysis of the datasets, while the database one has the script to upload 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 (c) or that analyses (a) a dataset
- *class* since this work could be divided into 2 different sections, Chain and Group, we used them as the main identifier to classify the files. There is also a generic class used when the computed data wasn't neither a chain nor a group.
- *aggregation* if the file is cathegorized by user or by page
- *name* the name of the metrics it computes
- *month* (optional) if the file is cathegorized by month
- *format* In create files the format of the output file is written directly in the file name in order to quickly understand what kind of 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 in order 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 is the folder structure of how the data was 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 is written in python but for some of the tasks we decided that using a bash script was a better idea, in particular in order to automatize processes like downloading the Wikimedia History Dumps or generating the datasets.

6 Conclusions

This is still an open project, so the results described in this report are not complete. The largest portion 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 living place, of the users, characterizes Wikipedia. We have seen only some languages but the study could be extended to all the available languages without any problem.

Future works will comprehend, besides the studies in different languages, an interactive dashboard with this data available online and one from 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 quickly neutralised. From the number of chains we can understand in which topic which people cannot reach an agreement; in Italy and south America these topics are sport, especially football, while 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.