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

Chapter 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.

Many articles means many users and therefore many potential conflicts. Avoiding these conflicts is the best way for this encyclopedia to grow.
Each Wikipedia page has four different sections:

- 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.

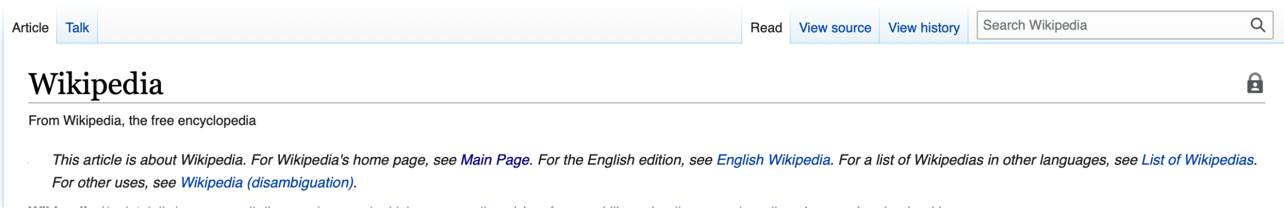


Figure 1.1: page structure

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.

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.

The results can be viewed in an interactive dashboard available online.

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 around 2010; 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.

Chapter 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).

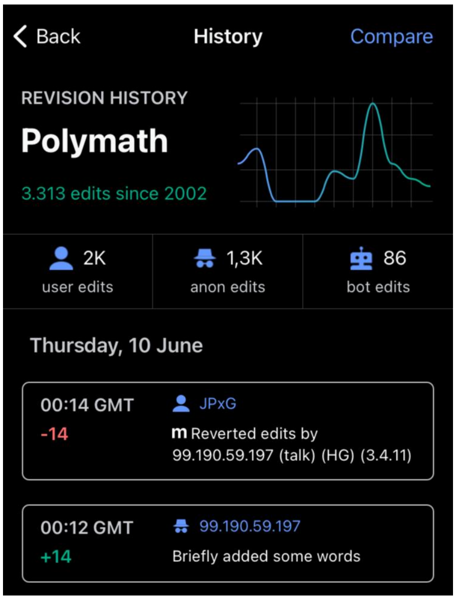


Figure 2.1: Mobile interactive visualization of the history

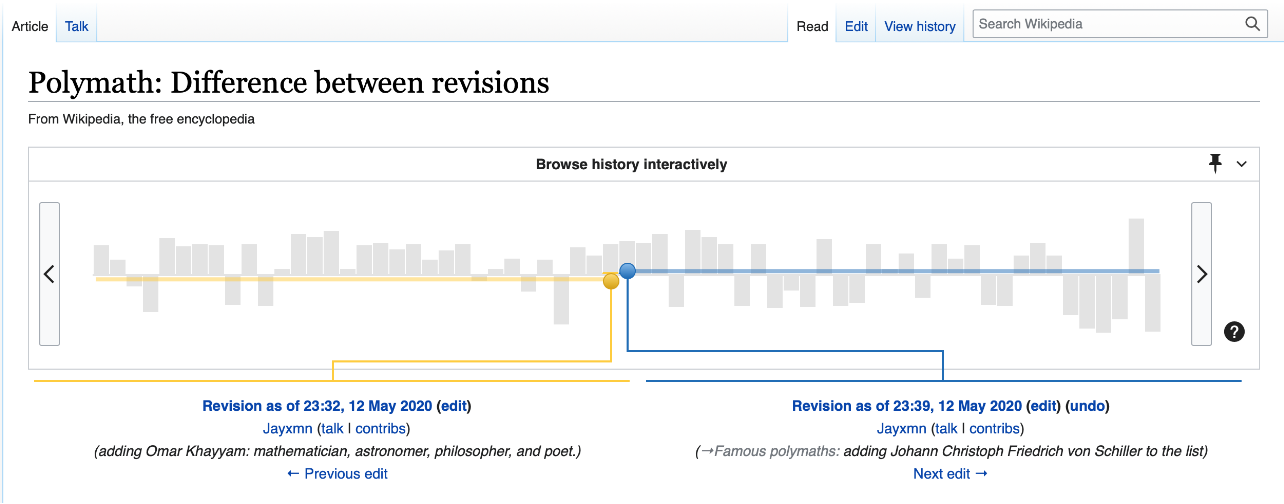


Figure 2.2: Interactive visualization of the history

2.2 Dataset

There are two datasets that store info about Wikipedia edits made available from the WikiMedia Foundation: a) the MediaWiki History and b) the MediaWiki History Dumps

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 interesting fields of each section are represented in the Tables 2.1, 2.2, 2.3. In the caption are present, 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	10479

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 because we are only interested in articles, i.e., the actual encyclopedia.

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 the 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.

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

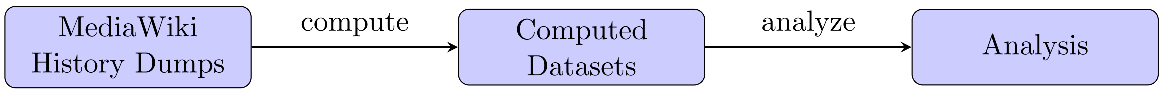
Controversiality G The controversiality G of an article is defined by summing the weights off all the chains there are on a page and multiplying by the total number of editors N involved in at least one chain.

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

Chapter 3

Methods

Considering the huge dimension 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, iIn this way a multilingual analysis on the most controversial topics can be conducted in different locations.



3.1 Computed Dataset

In the first skimming, only revisions that were involved in a revert were left. This dataset, whose schema is the same as the MediaWiki History Dumps, was sorted by both page and timestamp, and thanks to this screening, the size is now ~10% of the original. To achieve this we went through the compressed dataset line by line decompressing it on the fly and saving in a file only the entries we were interested in. Due to this the amount of ram required is very small and so is the disk space since all data is compressed. For the sorting part, we used Unix sort, which is the most optimized way to sort a file like this.

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

- Chains: in these datasets, the focus was on detecting revert chains in pages
- Group: in these datasets, the focus was on the number of reverts that users did or received based on the groups (admin, registered, anonymous).

3.1.1 Chains

The data about revert chains were computed from the compressed filtered dataset. Every time we used the filtered dataset we read it line by line, saving only the interesting information. The output is a JSON file, where each page corresponds a JSON object. For each page we save the list of chains and some statistics. A chain has a start and an end date, a list of revisions, and the users involved. This dataset is way smaller than the initial one so it is possible to browse the dataset in few seconds.

To identify a chain we used a function called *simple_chains* that differs from another called *complex_chains* for the fact that the first one to identify a chain of revert considers only contiguous reverts. We decided to use the simple one because we are only interested in chains that occur in a short time span since that is where most of the discussions take place. If more than 50% of users involved in a chain are bots the chain is not saved. There are two versions of this dataset, one that 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}
}
```

Regarding users, the object is very similar, but it is calculated differently. All the data we need is stored in the JSON pages. By analyzing that file you can extract all the chains in which a user has been involved and then calculate statistics in a similar way as for pages. Using this dataset allow us to compute this dataset 10 times faster.

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

The dataset was also calculated monthly for both users and pages, the schema is simpler than the JSON and this allows us to save it in a TSV using only one row for each month. Instead of saving all the data about the chain, we save the number of chains that are longer than 5,7,9. In Table 3.1 there is a sample page entry. To do this we have processed the JSON database one page (or user) at a time by dividing by month. We counted the chains per month basing on the start date of the chain.

title	**year_month**	**nchain**	**nrev**	**mean**	**longest**	**$\geq$ 5**	**$\geq$ 7**	**$\geq$ 9**	**G**
Loligo_vulgaris	2020-10	1	15	3.0	3	0	0	0	0

Table 3.1: entry of the mothly tsv

3.1.2 Group

Another interesting part of the study was focusing on the category a user belongs. Thanks to this we are able to track the habits of the users allowing us to understand, for example, if someone stops editing Wikipedia after several reverts from admins. Detecting these kinds of patterns is useful for community health: a user can be warned if its behavior could lead to a drop-off. 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 and gives us 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 users involved are a registered one and an admin, the order 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 anon we saved the number of reverts that were made by both anonymous (*anon*) and not anonymous (*not_anon*).

To calculate these metrics we use simple variables that are incremented as you scroll through the filtered dataset and initialized each time a new page is 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	pagina	2020-10	13	12	42	0	0	0

Table 3.2: entry of the revert page tsv

In the case of mutual reverts the procedure is similar but a bit more complex because we need to store the information of the whole page in order to correctly detect all the mutual reverts. The most efficient way to save such information is the use of dictionaries where we saved for each reverter the list of users who reverted and then at page processing time we computed a list of pairs that were used to calculate the other metrics.

id	page	year_month	mut_aa	mut_ra	mut_rr	anon	not_anon
1	pagina	2020-10	13	12	42	0	0

Table 3.3: entry of the mutual 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, this allowed us to save for each user-month the number of reverts made and received divided by group.

In this case, the dataset is browsed and processed month by month. When a user performs a revert the dataset gives us the id of the revision it has reverted but not the id of the user it has reverted. To solve this problem so 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 dictionaties 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 page tsv

The mutual revert analysis was more difficult to implement because to save the information about mutual reverts we need the dataset sorted by pages, but to get the data by user we should use the one sorted by timestamp. We solved this problem by storing the user-page-month in the dataset, so the information about the mutual returns of a user involved 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 the dataset by grouping by user or by 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	pagina	2020-10	13	12	4

Table 3.5: entry of the mutual page tsv

Chapter 4

Results and Discussion

The second step of this work was the analysis of the dataset just generated. Thanks to the structure and the heavy pruning analyzing these datasets is fast, This allows us to have a better workflow without interruptions. We analyzed the data in two ways: a descriptive statistic and an interactive one.

Descriptive For each dataset there is a script that runs and 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 month by month. Rankings instead are used to see in a more specific way the pages/users ordered by one of the metrics previously computed.

Interactive We decided to make available online an interactive dashboard. 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 Chains

From the analysis of the chains we can have an overview of an entire Wikipedia in a language, discovering which is the mean lenght of chains or the longest chain. Another aspect worth investigating is the relationship between alone reverts and reverts that are in a chain: more reverts in chains means more discussions, in this cases we could combine the data of other team members who analyzed the talk pages. While the pages chain are useful to have a less specific but wider view of the phenomenon, the users chain let us see if a specific user is involved in many chains and in which page is more active: in this sense we can define category of users: the ones who are active just in some topic or the other who reverts an all wikipedia. More interesting are the metrics by month, we can plot the trend of reverts in a page and see if it is always controversial or just in a specific storic moment related to something happened in the world. Plotting the metrics year by year allow us to understand the global activity of the users on wikipedia. Regarding users, we can define the lifecycle of a users and see when is more active and combining the data with the other team members we can say if its decrese of revisions it is related to a discussion.

4.1.1 Page

wars here the page ranked by the number of chains in italian and catalan

id	title	n_chains	title	n_chains
1	Serie A	195	Barcelona	68
2	Juventus FC	190	FC Barcelona	33
3	Matteo Renzi	179	Catalunya	30
4	AS Roma	176	País Valencià	26
5	Personale della WWE	167	Marc Márquez i Alentà	22
6	SSC Napoli	162	Mireia Belmonte i García	22
7	Inter	162	Girona	20
8	Roma	154	Rafael Nadal i Parera	19
9	Tiziano Ferro	141	Oriol Junqueras i Vies	17
10	Gianluigi Buffon	137	Català	16

Table 4.1: pages with more chains

let's see more specifically the italian first one, serie_A,

```
"title": Serie_A,
"revisions": [...]
"n_chains": 195,
"n_reverts_in_chains": 756,
"n_reverts": 5291,
"mean": 3.9,
"longest": 15,
"G": 2205218,
"M": 9479660,
"lunghezze": {"3": 96, "4": 66, "5": 15, "6": 11,"7": 2,"8": 3,"10": 1,"15": 1}
```

monthly let's see the trend of month by month of this page

4.1.2 User

wars

monthly

4.2 Group

From the analysis of the groups we can define different ranking of pages using the number of reverts of each group, or given a page we can plot the trend of the edits by group and detect the pages in which admins are more interested. It is possibile for each user to say if he is target of reverts from or if it is an admin reverted and the ration between reverted made and received. You can do so much analysis of this data, that is the reason why it is avaiable to everyone who needs it.

4.2.1 Page

reverts

mutual

4.2.2 User

reverts

mutual

Chapter 5

Infrastacture

All the code in available on Github `https://github.com/WikiCommunityHealth/wikimedia-revert`, there is an onrganiziation called WikiCommunityHealth where each team member commits its contribution to the project. For this dataprocessing we used python since is the best option to handle this amount of data. The data is currently stored in the Unitn servers of Cricca group.

Multi Language All the dataset computed are the results of several python scripts launched singularly. All the work has been done using the italian wikipedia as example. Automatizing the process allow us to run all the scripts in different languages. For achieving this automatation we used a bash script which 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 generate the dataset in the right place. the only requirements is that the dump is already been downloaded.

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 process like downloading the Wikimedia History Dumps or the generation of

style wide use of dictionary

Chapter 6

Conclusions

This is still an open project, further exploration will be done

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.

Allegato A Titolo primo allegato

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.

A.1 Titolo

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.

A.1.1 Sottotitolo

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.

Allegato B Titolo secondo allegato

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.

B.1 Titolo

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.

B.1.1 Sottotitolo

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.