



Universiteit
Leiden
The Netherlands

Opleiding Informatica

Scraping Scratch:
A Dataset of Comments and their Sentiment

Dyon van der Ende

Supervisor:
Fenia Aivaloglou

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

30/06/2020

Abstract

The online platform Scratch is a place where young children experience programming in a fun and playful way. Users can publish their programs and interact with others using comments. This interaction makes it an interesting subject for research. In this thesis, a database containing over 1,4 million comments from more than 199 thousand Scratch projects is created. All comments are then labelled with the language that they are written in and analysed for sentiment. All the meta-data that was collected during the process is also added to the database.

The analyses of data and meta-data revealed that commenting is the most used type of user interaction on a Scratch project and that newly created projects receive more comments than outdated projects. Furthermore, it showed that the majority of comments that were posted are written in English and that negative sentiment is less common than positive or neutral sentiment. The database and all the scripts that were used for this research are made publicly available, enabling further research in the field of Scratch, computer education and text mining.

Contents

1	Introduction	1
2	Related Works	3
2.1	Sentiment Analysis	3
2.2	GitHub	4
3	Methods	5
3.1	Data Acquisition	5
3.1.1	Project Scraper	5
3.1.2	Comment Scraper	5
3.1.3	Reply Scraper	6
3.2	Data Analysis	6
3.2.1	Language Detection	6
3.2.2	Sentiment Analysis	7
3.3	The Database	7
4	Results	10
4.1	Projects	10
4.2	Comments & Replies	10
4.3	Language	14
4.4	Sentiment	16
5	Discussion	18
6	Limitations	20
7	Conclusion	21
7.1	Future Work	21
	References	23
A		24

1 Introduction

Scratch is an online platform that was created in 2007, where users can create and share interactive programs and games, that are build in a visual block-based environment [11]. The code in Figure 1 is a complete program that launches a simple firework missile that explodes when the player clicks with the mouse. Because of this colourful and visual style of programming, it is accessible for people who are new to programming and especially younger children. This is one of the reasons why it is often used in computer education in schools and it is reflected in the age of the user group, that consists largely of users between the age of 8 and 19¹.



Figure 1: The code of “Fireworks display” by user ‘Dangerousgame’, which starts fireworks on a click.

Besides creating the projects and sharing them on the users public page, there are ways to engage with other users via loves, favorites, remixes and comments. Loves and favorites are comparable to the functionality offered by other social platforms such as Facebook, where a favorite expresses that it is more special than a love.

A remix allows users to use the code of another project for their own project. With a single click on the remix button of a project, the code of that project is opened in the editor of the user that wants to create the remix. The user can then change the code and publish it as a remix, while still giving credit to the original author. It is required that the code is actually changed before it can be published as remix.

A comment can be compared to the way that users can post reactions on a video on YouTube. It is possible to comment on another comment but only up to one level deep, see Figure 2.

On the platform there are rules in place to keep it safe for young users. One of the measures is a bad-language filter. This filters out the reactions that use blacklisted words². Another way is the reporting of inappropriate reactions by users³. All these metrics together with the number of views, are shown on the page of the project, providing insight to the popularity of a project, see Figure 3.

The social aspect of being able to engage with other users forms an important aspect of Scratch. Commenting is a way of connecting with other users and sharing feedback from which users can learn. Although positive impact of online feedback on the learning process has been found, there has not been a lot of research that has shown that this happens on a large scale for Scratch specifically. [4]

¹<https://scratch.mit.edu/statistics/>

²<https://en.scratch-wiki.info/wiki/Censor>

³[https://en.scratch-wiki.info/wiki/Comment_\(website_feature\)](https://en.scratch-wiki.info/wiki/Comment_(website_feature))

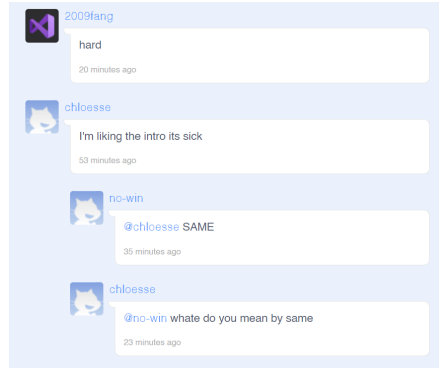


Figure 2: A few of the comments and replies from the comment section from the project in Figure 3. The replies from users ‘no-win’ and ‘chloesse’ are only one level deep, as it is the maximum depth for replies.

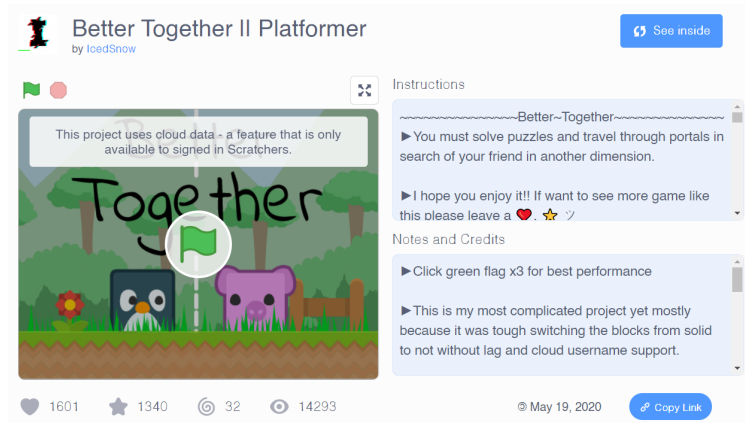


Figure 3: The project page of the game “Better Together || Platformer” by user ‘IcedSnow’. In the left bottom the number of loves, favorites, remixes and views are shown.

The aim of this thesis is to facilitate such research by creating a data set of comments and replies on user projects on Scratch. Each reaction will be scored for sentiment with SentiStrength and the data will be made publicly available. In this thesis the term *comment* refers only to upper level of reactions and the term *reply* is used to refer to a reaction that is on the second level, a reaction on a comment thus. When both types of reactions are meant, it will be stated explicitly. Also all comments and replies referred to are on user projects. There is the possibility for users to react to a user profile or a studio profile, but these were left out of the scope of this thesis, opting to focus on the project level reactions.

A sentiment analysis of Scratch comments has already been conducted on a small scale before by Fields et al. [4]. An overview of this research and other relevant works can be found in Section 2. Next, in Section 3 the process of scraping, analysing and compilation of the database are described. Then in Section 4 a data exploration of the database that contains over 770.000 comments and 700.000 replies from almost 200.000 projects, is done. This thesis ends with the discussion in Section 5, limitations in Section 6 and the conclusion along with recommendations for future work in Section 7.

2 Related Works

A number of papers have had a similar focus on creating a data set or analysing Scratch comments or projects as this research.

In a 2015 paper, Scratch comments were analysed for quality and their constructive, emotional and functional use. Drawing 8.000 random samples from over 5.000 different users, that were posted from January through March 2012. Of these comments, 2.273 were comments on projects and were used for the analysis. They found that that 58% of the comments had a more detailed nature, which they describe as “open new possibilities for interaction and action.” The other 42% were simple comments, meaning that they do not open the possibility for new (inter)action. The Sentiment analysis found that the majority of comments (72%) were positive of tone. Furthermore, they found that most comments could be classified as ‘motivational feedback’ or were used to build a following on the website. All comments were manually labelled, which was possible, because the number of comments was limited to 8.000. The comments that were used for this research were taken from an older data set, and because of this they analysed comments that were already a bit outdated at the time. [4]

In 2017, Hill and Monroy-Hernández published the most comprehensive data set of Scratch data that is available yet. The data set contains information of over 1 million users, 2 million projects, 10 million comments and source code of the different versions of Scratch that were used over time. All this data is from 2007 through 2012 and made publicly available. [7]

A data set of more recent Scratch projects was created by Aivaloglou et al.. Every project in this data set was graded by Dr. Scratch, a tool that scores Scratch projects based on level of computational thinking. They did this with over 250.000 projects and the grades can also be found in the database. The goal of the research was to facilitate further research in the field of software engineering and computing education. The data in this research was not randomly collected, but sorted by newest projects, resulting in the most recent data that is publicly available. [1]

2.1 Sentiment Analysis

The term sentiment analysis is often used interchangeably with the term opinion mining, because they denote the same field of study according to Pang and Lee [10]. Both are part of the field of text mining, which has gained a lot of interest in recent years, as more and more textual data became available via the internet [6].

The Oxford Handbook of Computational Linguistics describes sentiment analysis as “. . . one component of subjectivity analysis. Technically, it refers to the task of identifying the valence — positive or negative — of a snippet of text. The identification can be done at a wide variety of granularities, from a word type — either in or out of context — to a phrase, sentence, paragraph, or entire document. [. . .] At any granularity, the task can be a simple binary one (positive vs. negative) or an ordinal one (e.g. 1, 2, 3, 4, or 5 stars).” [3]

Research in the field of psychology, however, has shown that people can express two emotions at the same time [2]. Because of this, the sentiment analysis tool SentiStrength gives two scores to a text: a negative sentiment score and a positive sentiment score. The tool is especially targeted towards short texts on social media and can achieve a great level of accuracy. It works with a lexicon based approach, where a dictionary (lexicon) is kept that stores sentimental information of a word, e.g. the word *happy* expresses positive sentiment, while *angry* is negative. Next to this lexicon, SentiStrength also has an idiom list, booster word list, negation word list and an emoticon

list. These provide additional information that can improve or adjust the meaning of the words in the lexicon. [15]

But as Liu notes, there are some downsides to this approach. The meaning of words heavily depends on the context. For example, the meaning of the word *sucks* is completely different in the following two sentences: “this camera sucks” versus “this vacuum cleaner really sucks” [8]. Besides this, irony and sarcasm are also a weakness of SentiStrength and the lexicon based approach in general [16].

Because of the options for engagement on Scratch, it can be seen as a social media platform. In the field of sentiment analysis, a lot of research has focused on social media and most notably Twitter, because of its textual nature and broad usage worldwide [14]. In the 2017 run of SemEval, 48 teams participated in the task of sentiment analysis on Twitter. The goal for the teams was to build the most accurate sentiment classifier for tweets with the most state of the art techniques, the most promising being machine learning [13]. Another interesting way that Twitter is used for in the field of sentiment analysis is by Pak and Paroubek, who use Twitter to create a corpus with positive and negative sentiments. The method that is described uses emoticons to find positive and negative tweets and then train a classifier with the words in the tweet, removing the need human labelled data completely [9].

2.2 GitHub

As Scratch hosts the code of millions of projects from millions of users, it is in some way more comparable to GitHub⁴ than it may seem initially. Any GitHub user can create a repository, which will host the files of a (software engineering) project. Contributors of a project that are allowed to edit files, can upload new versions, so called commits. Other contributors are then able to leave a comment on a commit, just as Scratch users can comment on a Scratch project.

Guzman et al. analysed the sentiment of the commit comments on GitHub with SentiStrength to see if the sentiment was related to the programming language of a project, the weekday of posting or the geographical distribution of the team. They found that comments on projects written in Java are generally more negative of sentiment, just like comments that were posted on Monday. A relation between geographical relation and sentiment could not be found. These results are very interesting, but the the research was not representative enough to draw any real conclusions from this, having only analysed 90 of the most popular GitHub projects. [5]

Because of the technical nature of the GitHub platform, it can be difficult to correctly analyse comments, as they for example can contain code or technical descriptions. To address this, Rishi developed a new approach for mining sentiment on GitHub, using a method from the field of sociology that is used to model the interaction of small groups. This approach is then used with machine learning to create a model that performs better than SentiStrength in labelling sentiment on GitHub. [12]

⁴<https://github.com/>

3 Methods

The process of scraping and analysing the comments has been divided over several tools that are run in the sequence described in this section and were created for the purpose of this thesis. The modularity of the tools made it easier to redo a small step in the process, instead of having to restart the whole process over again when something had to be adjusted, which can be a costly operation when working with large quantities of data. All the tools are written in Python.

As a starting point, a data set that was created by Zeevaarders was used to compile a list of project ids from [18]. This data set contains around 7 million projects spanning from 2007 through October 2019. This saved a lot of effort on collecting project ids.

As the data contains a lot of projects, some of which dated, it would take a lot of time to scrape all projects. Therefore, it was decided to scrape 200,000 projects from 2019. The projects were sorted by author, so that it gives a more complete overview of a user. The most recent projects in the data are from 28th of October 2019.

All data that was collected can be found at <https://github.com/dyonende/Scratch-scraper> along with the code of scripts that are described in this section.

3.1 Data Acquisition

The data was acquired by scraping Scratch. This phase was divided into three different steps and took several days to complete.

3.1.1 Project Scraper

The project scraper (`projectscraper.py`) takes a list of project ids as input file, where each id is on a separate line. With this id the URL can be constructed to download a JSON file containing all information of a project. This step is necessary, as for the scraping of comments, the `username` and project id are required and the existing database does not provide the username. The URL that is constructed is `https://api.scratch.mit.edu/projects/{projectID}`, where `{projectID}` should be replaced with the project id. The JSON on this location is downloaded with the python request library and then parsed with the json library and printed in CSV format to `stdout`, resulting in a CSV of project information. The output is redirected via the command line to a new file. From this file the username and project id can be extracted, which is required for the next step. After the scraper was run, 195,563 projects were successfully scraped and parsed, meaning the other 4,437 projects had failed.

3.1.2 Comment Scraper

The next script that is used is `commentscraper.py`. This takes a file with on each line the username, the project id and the number of comments and replies on a project as argument. These arguments are extracted from the output of the previous step. If a project has no comments and replies, it will be skipped by the program. Of all the projects that were scraped, only 88,927 projects had comments.

Then, a new URL will be constructed of the form `https://api.scratch.mit.edu/users/{username}/projects/{projectID}/comments?offset={offset}&limit={stepsize}`. Due to limitations on the side of the Scratch servers, it is only possible to access a maximum of 40 comments per request. To address this, a for-loop cycles with steps of 40 through the `offset` parameter, each

time requesting the next 40 comments, all in a JSON file, until an empty file is returned and it will move on to the next project. Each file is again parsed and printed to `stdout` in CSV format. The output of this program is thus a CSV of all comments, which will be used for the next steps. From the 88.927 projects only 87.796 projects were downloaded and parsed correctly, whilst the others failed. This resulted in 772.322 comments.

3.1.3 Reply Scraper

After inspecting the comments from the previous step, it was noticed that none of the comments had a parent comment, which indicated to which comment the comment is a reply to. This seemed not right and upon further inspection, it turned out that the way Scratch has structured the replies, is that it loads the replies separately from the comments, even though the replies are still comments and the JSON file contains the same attributes. This means that for each comment that is loaded, a request has to be made to ask for replies on this comment. This is feasible for the website, as it loads only 20 comments at the time. However, to scrape the replies, another tool is needed that is very similar to the comment scraper. The new URL is of the form `https://api.scratch.mit.edu/users/{username}/projects/{projectID}/comments/{commentID}/replies?offset={offset}&limit={stepsize}` and the input of the program is a text file with `project_author_username project_id comment_id` per line. This data was extracted from the output of the comment scraper. Eventually this resulted in 718.158 replies.

The project scraper was started on 8th of April 2020 and finished two days later on the 10th. The comment scraper was run directly after the project scraper was finished and ran for two days as well, finishing on the 12th of April. The reply scraper was started on the 4th and finished on 8th of May 2020.

3.2 Data Analysis

After the data acquisition was finished, it was time for the sentiment analyses, but before that, the language that the comments were written in, had to be detected.

3.2.1 Language Detection

In order to analyse the comments for sentiment, their language needs to be identified, since SentiStrength works in English only with the version that was used. The language identification is done with the script `languanote.py`. It takes the file that was created by the comment scraper and adds two additional columns: `language` and `isReliable`, indicating the language and a Boolean that tells if the prediction is reliable. Each comment is analysed by `pycld2`⁵, a python fork of Compact Language Detector 2⁶. This can detect the language fast and with a high level of accuracy [17], but only if texts are not too short. To address this, a simple fix is used: if `pycld2` can not detect the language and the sentence consists of up to 3 words, the program will check if all words are in the `nlTK.corpus` word list, a list of English words. With this method, single words and simple sentences, such as “wow” and “cool” can still be detected as English. It took about 10 hours to label all comments and another 10 hours for the replies.

⁵<https://pypi.org/project/pycld2/>

⁶<https://github.com/CLD20wners/cld2>

3.2.2 Sentiment Analysis

The tool that was used for the sentiment analysis was SentiStrength [16]. The free version was used with the standard settings, but it performed the analysis impressively fast. To rate all comments and replies took less than a minute in total. Manual inspection of some of the results showed that the language detector often did not recognise a text as English, even though it was written in English. Because of this, it was decided to do the sentiment analysis on all comments and replies, even if they were not labelled as English.

3.3 The Database

From the output data of the previous steps, the ones from the project scraper, reply scraper, comment scraper and the sentiment analysis are added as tables in a sqlite3 database. The database consists therefore of five tables: `Projects`, `Comments`, `Replies`, `Comment_Sentiment` and `Reply_Sentiment` and all tables are encoded in UTF-8. In Table 1, 2, 3 and 4 the columns of the tables are listed with a description.

Some corrupt entries, most often caused by uncommon characters that confused the import tool, are removed. This was done on manual inspection of the data. The irrelevant columns, `visible`, `reliable`, are removed from the `Comments`, `Replies` and `Projects` table as they add no value: every project and comment is visible, otherwise it wouldn't have been downloaded; the language is only labelled as unreliable when it could not be identified.

The `Projects` table contains all the information of the projects that was provided in the JSON file at time of downloading. The `Comments` table also contains all the info from the JSON file, with the added language column, which holds the language that was detected by the language identifier.

Note that there was some time between starting the project scraper and starting the comment scraper, as this may cause a small difference between the number of comments stated in the `Projects` table and the actual number of comments in the `Comments` table.

The table with the `Replies` is almost the same as the `Comments` table, but it has two extra columns: `commentee_id`, `comment_parent`.

Field	Description
author_id	The unique integer number assigned to a user
author_username	Username string that the user has chosen
project_id	A unique integer number assigned to a project [PRIMARY KEY]
title	The title string that a user has chosen to name it's project
date_created	The date on which the project was first created (DD-MM-YYYY HH:MM)
date_modified	The date on which the project was last modified (DD-MM-YYYY HH:MM)
date_shared	The date on which the project was made public (DD-MM-YYYY HH:MM)
views	Number of views on a project
loves	Number of loves on a project
favorites	Number of favorites on a project
comments	Number of comments on a project
remixes	Number of remixes of a project

Table 1: Description of the Projects table.

Field	Description
project_id	A unique integer number assigned to the project commented on [FOREIGN KEY: Projects.project_id]
project_author	Username string of the author
comment_author_id	The unique integer number assigned to the user that posted the comment (comment author)
comment_author_username	Username string of the author of the comment
comment_id	A unique integer number assigned to each comment [PRIMARY KEY]
comment_parent	The comment_id of comment this comment is a reply to (always NULL as this table contains no replies)
commentee_id	The unique integer number of the user that posted the comment this comment is a reply to (always NULL as this table contains no replies)
comment_date_created	Date when the comment was created (DD-MM-YYYY HH:MM)
comment_date_modified	Date when the comment was modified (DD-MM-YYYY HH:MM)
comment_content	The text that the user wrote as a comment on the project
language	ISO 639-1 2-letter abbreviation of the language that the comment is written in. 'un' means unknown

Table 2: Description of the Comments table.

Field	Description
project_id	A unique integer number assigned to the project commented on [FOREIGN KEY: Projects.project_id]
project_author	Username string of the author
comment_author_id	The unique integer number assigned to the user that posted the comment (comment author)
comment_author_username	Username string of the author of the comment
comment_id	A unique integer number assigned to each comment [PRIMARY KEY]
comment_parent	The comment_id of the comment this comment is a reply to [FOREIGN KEY: Comments.comment_id]
commentee_id	The unique integer number of the user that posted the comment this comment is a reply to
comment_date_created	Date when the comment was created (DD-MM-YYYY HH:MM)
comment_date_modified	Date when the comment was modified (DD-MM-YYYY HH:MM)
comment_content	The text that the user wrote as a comment on the project
language	ISO 639-1 2-letter abbreviation of the language that the comment is written in. 'un' means unknown

Table 3: Description of the Replies table.

Field	Description
comment_id	A unique integer number assigned to each comment [FOREIGN KEY: Comments/Replies.comment_id]
positive	A number in range 1 to 5 indicating how positive the comment is (1=neutral,..., 5=extremely positive)
negative	A number in range -1 to -5 indicating how negative the comment is (-1=neutral,..., -5=extremely negative)

Table 4: The structure of both the Comment_Sentiment and the Reply_Sentiment table.

4 Results

In total the data contains information of 199.552 projects, 772.289 comments and 707.669 replies (1.479.958 in total), about 0,37% of all projects and 0,55% of all comments and replies shared on Scratch⁷. The comments and replies are posted on 87.793 different projects, leaving 107.759 of the projects without a comment or reply.

4.1 Projects

Comments, favorites, loves and remixes can be seen as interaction or engagement on a project. If these are divided by the number of views a project has, it gives a sense of how the interaction is spread on Scratch, which is shown in Figure 4. It is immediately clear that the mean, median, mode and third quartile are really low, all scoring below 0,50, especially for remixes, where the mean is the only value that is higher than zero. A large number of projects get no engagement or very little engagement. In Table 5 the percentages of how many projects did not get a form of engagement are found. It shows that one third of all projects did not get any form of engagement. But there are also a lot of outliers that receive a lot of engagement. This is reflected in the fact that the mean is higher than the mode and median for all metrics. In Appendix A some outliers are listed in Table 9 and Table 10.

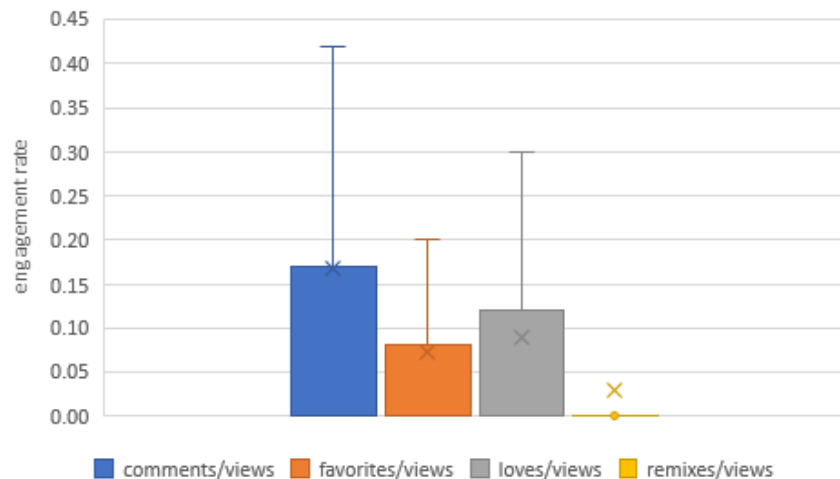


Figure 4: Project engagement statistics. Comments are the total number of comments and replies combined. All values are divided by the number of views of the project. Outliers are not displayed.

4.2 Comments & Replies

In Figure 5 a timeline shows when the projects in the data were created. An interesting observation regarding this, is that the drop in new projects from around September through October 28th can also be found back in Figure 6, which shows the number of new comments and replies. The drop in new projects is solely based on the selection of projects that were used for this research.

⁷According to the statistics on Scratch's website on 16th of May 2020.

	% of projects
No comments	55%
No favorites	56%
No loves	52%
No remixes	81%
No engagement	33%

Table 5: The percentage of projects in the data that did not receive a form of engagement.

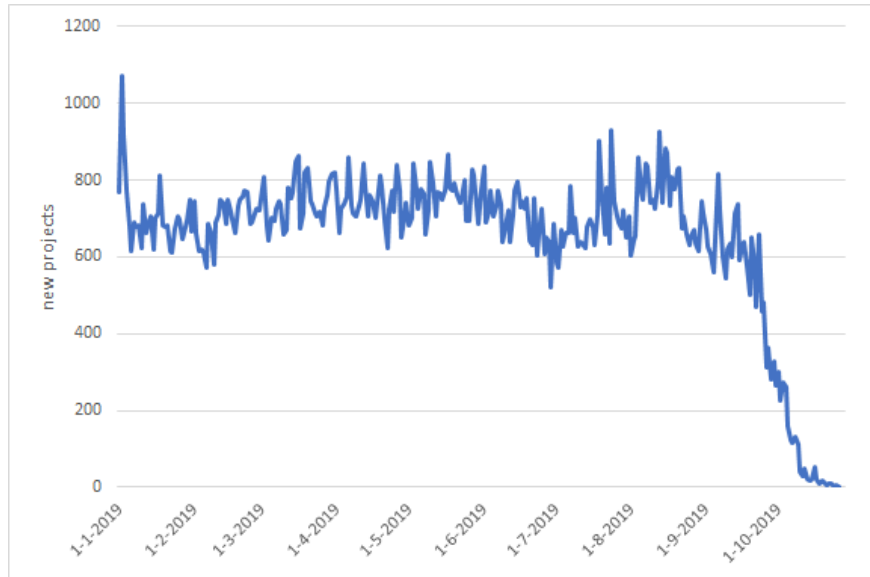


Figure 5: Timeline of new projects.

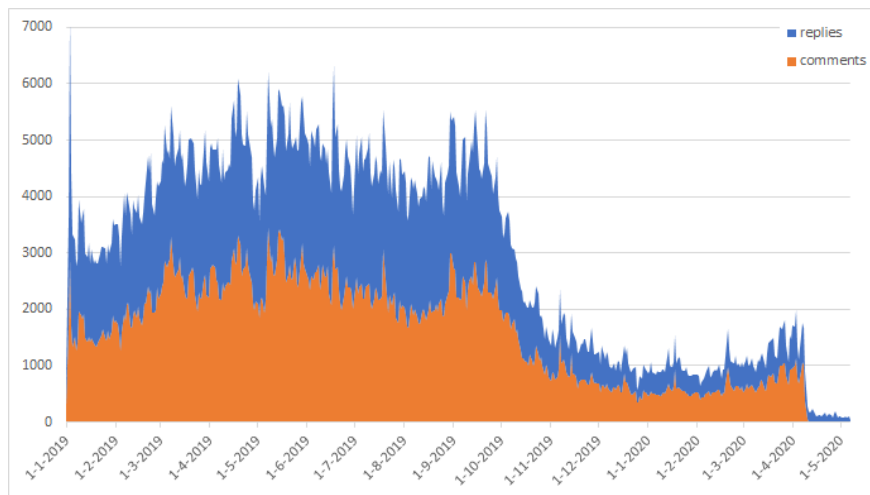


Figure 6: Timeline of new comments and replies.

Of all comments, 35,14% has at least one reply. From these comments, the number of replies is displayed in Figure 7. The most frequent value is 1 reply per comment, but there are also a lot of comments that have a far higher reply count. The comment with the most replies was by user ‘iceberg-the-icewing’ who commented on his own project (ID: 318474570):

“Caspian huffed ‘may be fun for you, but frankly, not for me’ he growled and suddenly had a few spears of water behind him” (ID: 117642522)

This comment received 1788 replies in total, almost twice as much as any other comment. The quote appears to be from a book by C.S. Lewis and the replies are quoting the rest of the story it seems.

A short list of popular comments is given in Table 11 in Appendix A.

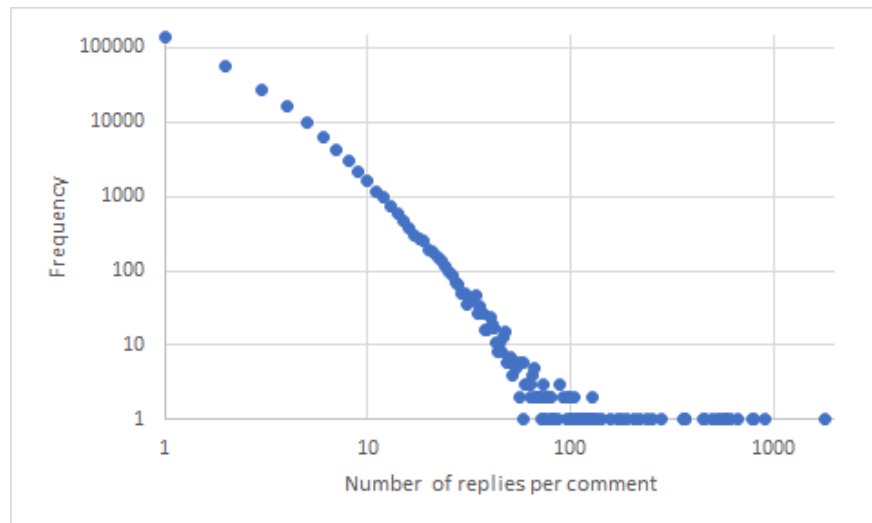


Figure 7: Replies on a comment. Only comments with replies are plotted.

Figure 8 and Figure 9 show how much characters and words are in a reaction. It clearly shows that there is a negative correlation between the number of characters and frequency. Most reactions are short of nature, because the most common reactions consist of either one or two words or emoticons. In Table 12 in Appendix A the most common reactions for comments and replies can be found.

Looking at the number of words, it is clear that almost all comments and replies are under a 100 words long (99,9%), again with some exceptions. Comparing the comments and the replies in both figures, it shows that there are more comments that contain between 10 and 100 characters, but outside of those bounds it is a mix. There are also more replies than comments, when they contain more than 30 words.

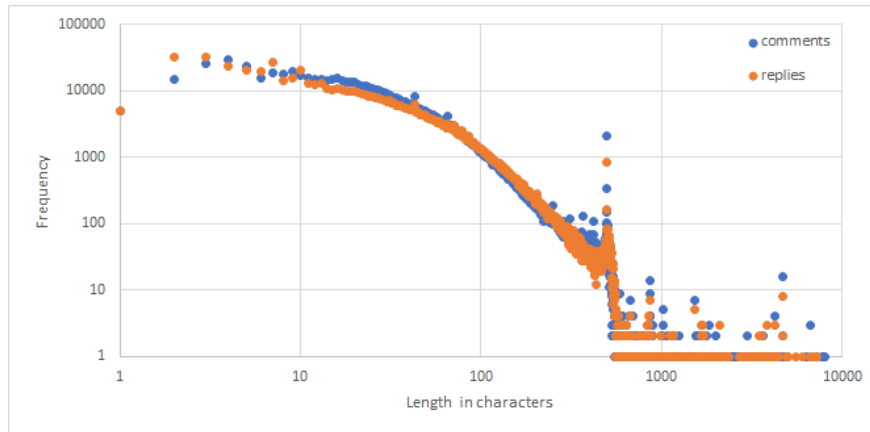


Figure 8: Number of characters per comment and reply.

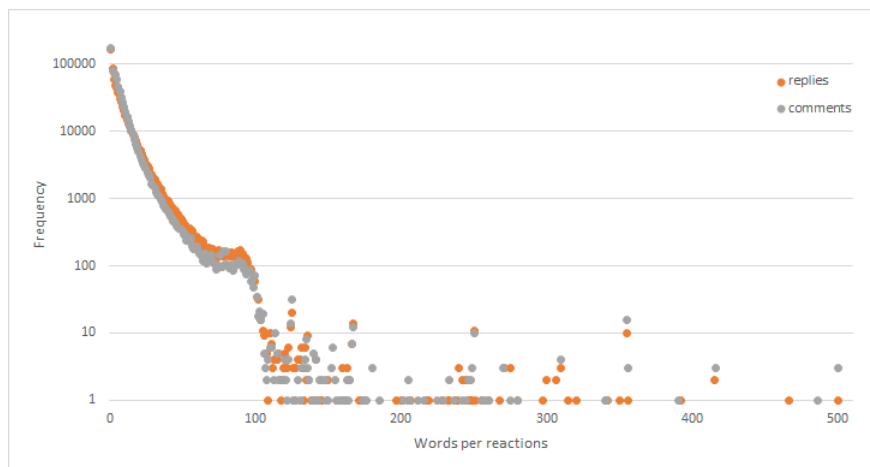


Figure 9: Number of words per comment and reply.

There is also an interesting difference between how often one user comments on the same project. This is shown in Figure 10, where the distinction between the author of the project and a different user is made. At around 20 comments there is an intersection between the two trends. This means that under 20 comments it is more often not the author that keeps commenting on a project, but over 20 it is the author more often.

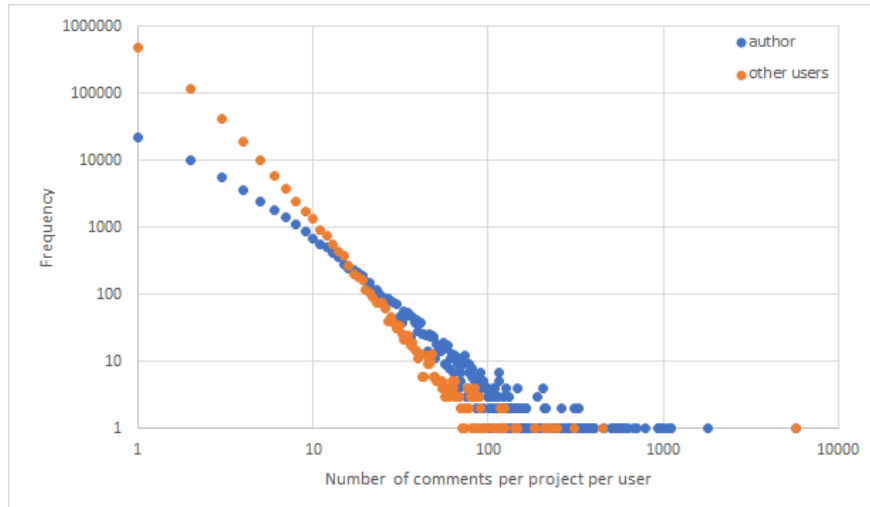


Figure 10: Number of times a user comments on a project. The distinction between the author of the project and a different user is made. Comments and replies are combined.

4.3 Language

In Table 6 the most frequently used languages of the comments and replies are listed. Almost all of them are identified as English (74%) or could not be identified (22%). If those that are unidentified are left out, 95% of the comments are in English, followed by Japanese, Korean and French. This is quite remarkable, especially if we compare this to were the users of Scratch are based, shown in Table 7. Here the largest English speaking countries account for a combined 52% of the users, notably less than the percentage of English comments and replies in the data. China is the third country on the list with 6% of the users, but this is not reflected in the language, where Chinese does not even account for 2% of the comments and replies if the English and unidentified comments are left out.

Language	Frequency	% of total	Unknown left out	English and unknown left out
English	1.098.334	74,214%	95,389%	-
Unknown	328.537	22,199%	-	-
Japanese	29.312	1,981%	2,546%	55,215%
Korean	3.010	0,203%	0,261%	5,670%
Scottish	2.549	0,172%	0,221%	4,802%
French	2.163	0,146%	0,188%	4,074%
Polish	2.022	0,137%	0,176%	3,809%
Danish	1.408	0,095%	0,122%	2,652%
Portuguese	1.276	0,086%	0,111%	2,404%
...
Chinese	762	0,052%	0,066%	1,437%

Table 6: Most used languages over all reactions (comments and replies combined). In the last two columns, the comments that were labelled as ‘unknown’ and ‘English’ or ‘unknown’ are left out respectively. The “...” indicates that some languages were left out for simplicity.

Country	Users	Percentage
United States	21.389.542	42%
United Kingdom	3.226.278	6%
China	3.045.203	6%
Australia	1.798.424	4%
Poland	1.795.426	3%
Spain	1.504.042	3%
Canada	1.440.608	3%
France	1.027.251	2%

Table 7: Official Scratch data of where their users are from (9th of May 2020).

4.4 Sentiment

In Figure 11 five heatmaps show what the sentiment of comments and replies is. Although most comments and replies are labelled as neutral (-1, 1), there is a clear positive tendency, especially in the comments. The most extreme comments can be found in Appendix A, Table 13, Table 14 and Table 15. Replies are also positive, but less pronounced. Even though it is not sure what language the unknown comments and replies are written in, the same pattern can be recognised as in the English comments. There seem to be more comments that are stronger negative than positive, when the comment expresses two rather strong sentiments, but this is only applicable for the English comments and replies.

Comparing these results with the findings of Fields et al. [4], there are some interesting differences. They find that most comments (72%) are positive of tone and that there are the same percentage of negative and neutral comments, 14%. Here, just taking the English comments and replies, 41,35% of them are neutral (-1, 1), 48,89% are positive (positive > 1) and 17,53% are negative (negative < -1). Together that is more than 100%, since comments can have two sentiments. But as is clear, Fields et al. numbers differ. The difference in negative reactions is not large, the difference in positive and neutral reactions is, however, quite big. By converting the sentiment scores to a single label, having the strongest sentiment count and equal sentiment as neutral (even though it is not), the new percentages would be: negative 12,02%, positive 42,88% and neutral 45,10%, see Table 8. Still the results aren't quite comparable, having still a higher percentage neutral comments and lower percentage positive comments.

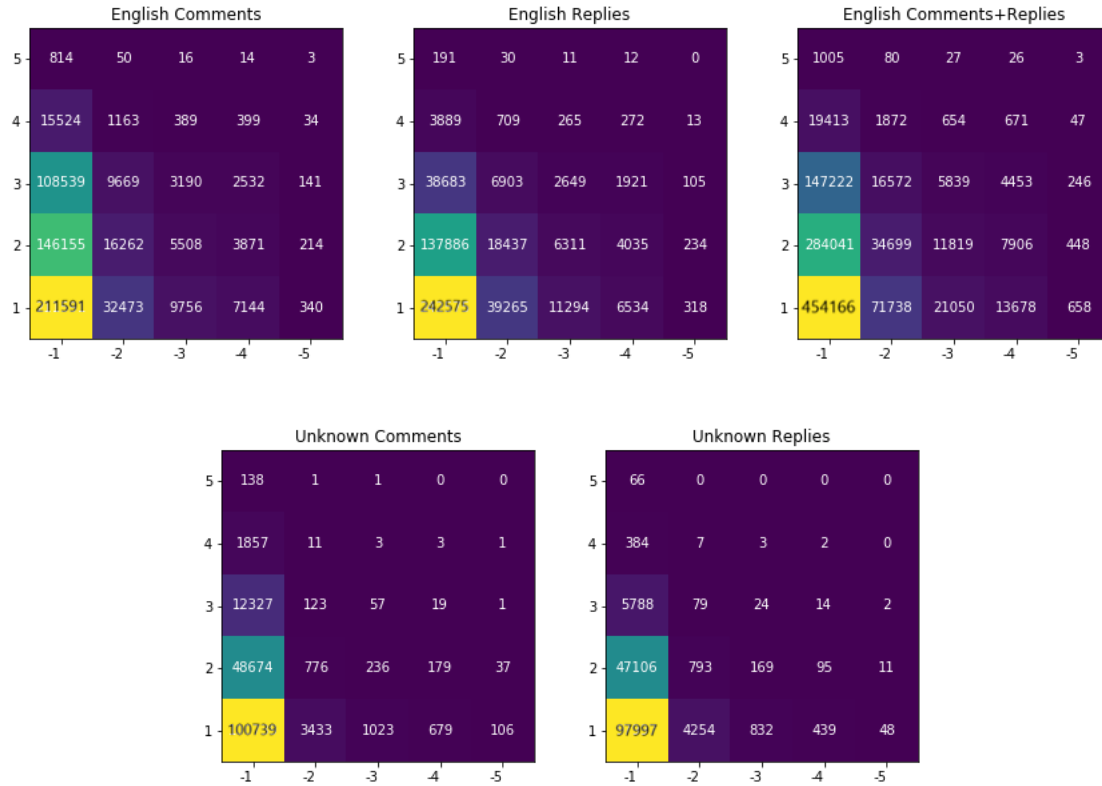


Figure 11: Heatmap of the sentiment. The X-axis shows negative sentiment, Y-axis shows positive sentiment. Unknown means that no language could be identified. Brighter colours means a higher frequency of the sentiment.

	Positive %	Neutral %	Negative %
Original 2-scale sentiment	48,89	41,35	17,53
Converted to one label	42,88	45,10	12,02
Fields et al. [4]	72	14	14


Table 8: Comparison with [4] and the English data from the data set.

5 Discussion

In Section 4 the data of almost 200 thousand projects and over 1,4 million comments and replies was analysed. It was found that the amount of remixes per view on a project are significantly lower than the other types of interaction. This makes sense as the amount of effort that is required to create a remix is much greater than to create a comment, favorite or love. Commenting is the most common, which can be contributed to the fact that it is possible to post multiple reactions per project, whereas a love or favorite can only be expressed once.

Then, by comparing the amount of new projects and new comments and replies per day, a drop in frequency was found in both statics. This drop in new comments and replies is likely caused by the fact that most comments and replies are posted on projects that are relatively new. The data showed that the majority of projects have no or not many comments, but that there are exceptions that receive a lot of comments and replies. This explains the high number of reactions overall and confirms the fact that most projects and comments don't receive a lot of interaction, but there are some exceptions that receive a lot of interaction, the 'viral' ones.

Focusing more on the content of the reactions, it appeared that comments and replies tend to be relatively short of nature. However, a spike is noted at (around) 500 characters per reaction. This is because there is a character limit of 500 per comment. But this does not explain why there are still comments that have over 500 characters. This is caused by the fact that the comments are encoded with HTML entities which, for example, escape the '&' character to '&', creating four more characters. If a comment contains 500 ampersands, it will result in 2.500 characters.

A similar thing happens with the number of words per comment. More than 250 words should technically not be possible, as words are separated with a space and a word is at least 1 character long. Scratch, however, allows a user to embed some pictures in a post. These pictures are a link pointing to the location of the picture. If users use only pictures in their messages, it will quickly contain more words, as the links are of the format ``. So again, this has to do with the way the comments are encoded. The maximum of 500 words can then be explained, because to include an image, a string of with an underscore as prefix and postfix is used. The length of this string counts for the character limit. So the string `_:D_` (that shows the emoticon ) fits exactly 100 times, creating exactly 500 words.

The languages that were spoken in the comments were compared to the location of the user base of Scratch. This showed that even though user that are from English speaking countries account for about half of the user group, the number of English comments and replies was a lot higher. The location of each user can be found on the user page, therefore it would be possible to inspect this aspect further. Unfortunately, this was out of the scope of the thesis, as it would have taken up too much resources.

Another explanation for the skew towards English, is that the language detector seems to have trouble recognising some languages, but this seems unlikely. Comments and replies that are labelled as English, or languages that use their own set of characters are usually recognised correctly. It is not likely that the language detection failed to miss a lot of comments and replies in Chinese, as the language uses its own character set. That could also explain why Japanese and Korean have the highest frequency after English: the unique character set makes it easy to recognise the language. This is in contrast with English, that only uses ASCII characters in normal usage, just as a lot of other languages. That could also explain why Scottish has such a high percentage, it's because it has some resemblance to the English language. This holds for reactions labelled as 'unknown' as well. Most of those comments and replies are in fact in English, but sometimes use slang or contain

grammatical errors, leading the language detection to fail. An example of a comment that is labelled as unknown but is actually in English is:

“M i n e 1 0 0 % i s n o t E n j o l r a a r e l a t a b l e m e m e s . . .))”

by user Jolllly. It says “Mine 100% is not Enjolraa relatable memes...))”, but with a space between every character. A human could recognise this easily and make out that it is indeed English. The language detector is not as advanced. So after manual inspection of the comments it seems like most (if not all) comments and replies that are labelled as Scottish are in fact English and that we should consider them as such. Meaning this was a fault made by the language detector. A selection of these comments can be found in Table 16, Appendix A.

The fact that the comments contain HTML entities, does not contribute well to language detection. Lots of accents and characters that are common in certain languages (e.g. ñ and ç) are escaped. The language detection does not handle HTML entities and so it does not recognise words that contain them. The English language does not contain non-ASCII character usually, so it is not affected by this. A selection of comments that contain HTML entities can be found at Table 17, Appendix A.

Because the language label is not always correct, it was chosen to still analyse all comments for sentiment, as it may contain some valuable information. SentiStrength can handle HTML entities, so for the sentiment score it should not make any difference.

So, despite that the language detector is not perfect, it still seems most likely that English is used as the Lingua France of Scratch.

Looking at the results from the sentiment analysis, the sentiment of the comments and replies is overall more positive than negative. This was also in line with what Fields et al. found, but they found 72% positive sentiment and 14% neutral sentiment, while this data showed 43% positive sentiment and 45% neutral sentiment. This could be explained by the larger amount of data that was used for this study, but also by the fact that those comments were hand labelled. An explanation for the negative comments is, that in their research, comments could only have one label instead of 2 scores to show sentiment and as stated before, the positive sentiment is often stronger. Another explanation would be that the sentiment analysis tool is too strict and labels a text as neutral even though it is positive or negative. But as said in Section 1, there is a bad-language filter in place, which prevents some types of bad behaviour in reactions. This could affect the sentiment of comments and replies towards a more positive sentiment.

6 Limitations

During the process of collecting the data and writing the thesis, some limitations were noticed which will be discussed in this section.

As said in Section 1, the decision was made to only scrape comments and replies on project pages. These are not the only kind of reactions that are possible on Scratch. This was mainly a practical decision, as there was not enough time to scrape these types of reactions and analyse them as well. This argument is also why initially 200.000 projects were selected to be scraped. Because the tools were build to run on the output of each of the preceding step, it was not possible to let the scraper run for a certain amount of time and see how much data it would collect.

Another downside of the tools is that they were designed only for the purpose of this research in mind. This means that they can not be generally used for other websites or if Scratch changes the way the systems works, without having to modify the tools.

In this research the context of a comment or reply has not been taken into consideration. For replies especially, this can reveal a lot about the sentiment. For example, when one users replies that he or she agrees with another comment, the way that this is written can be neutral, but the meaning is that the users shares the sentiment of a comment. This is clearly illustrated in the example of Figure 2. It can be quite challenging to take context into consideration and this is not made easier by the way that Scratch handles replies. Since there is only one level deep of replies, it can be hard to find out for which comment or reply the reply that is being inspected refers to.

Lastly, there are also some remarks regarding the data. The comment count attribute in the `Projects` table can differ from the actual amount of comments and replies in the database. This is caused by the fact that, as already described, the project scraper was run first, which took more than a day, and then the comments and replies were scraped. In the meantime, comments and replies could have been added or deleted. Therefore, if an accurate count of comments and replies is required, this should always be counted from the `Comments` and `Replies` tables. And as is explained in Section 5, HTML entities are not decoded in the data, so for text mining purposes it is advised that these are decoded for optimal usage.

7 Conclusion

For this thesis, the data and meta-data of 199.552 projects created between 1st of January 2019 and 28th of October 2019 was scraped and parsed. Then all data and meta-data of the comments posted on those projects were also scraped and parsed, as well as all replies on comments. This resulted in 772.289 comments and 707.669 replies, accumulating to 0,55% of all comments on Scratch (16th of May 2020). By analysing the metrics for engagement, it was found that most projects receive little to no engagement from other users, but that there is a small percentage of projects that go viral and receive a lot of engagement.

The next step was to identify the language of all comments and replies and score them for positive and negative sentiment. This revealed that English is the most common language for reactions, and that the sentiment is overall more positive than negative.

Finally, all the data that was gathered in the process, was neatly ordered in a database. The data is made publicly available, together with the source code of the tools that were created to scrape the data. Both are available at <https://github.com/dyonende/Scratch-scraper>.

7.1 Future Work

Making the data and tools available publicly enables further research in the field of text mining, computer education, linguistics and Scratch. Here, some possible directions for future research are given.

Firstly, it would be interesting to analyse all non-English comments and replies with a sentiment analyser that is specific to the language, to see if there is a difference between English and non-English sentiment.

Secondly, an analysis on the effect that positive or negative reactions can have on a user, could be an important contribution to the use of Scratch in computer education. If this shows that positive reactions on a project can motivate the user to continue creating projects, it is likely that they become better at it, especially considering the fact that a large part of the reactions are positive of nature.

Thirdly, the effect of negative reactions could be measured. If a negative reaction is posted, it could trigger more negativity in the comment section. But also, if there is a specific group of users that is generally more negative than other users, maybe the platform could benefit from blocking them.

Fourthly, in the field of linguistics, it would be possible to do research on the language that Scratch users use. In comparison to Twitter and Facebook, the users of Scratch are much younger. Therefore, it allows research to see how the language of users develop. For example, the number of grammar errors could be tracked over time, to see if their use of language improves.

And lastly, the impact of the COVID-19 outbreak on the usage of Scratch could be explored. The number of new comments and replies has more than doubled since February 2020. With the tools that were developed, comments and replies from the start of the outbreak can be scraped and analysed and then compared to the data from before the outbreak, to see if the sentiment has changed.

References

- [1] E. Aivaloglou, F. Hermans, J. Moreno-León, and G. Robles. A dataset of scratch programs: Scraped, shaped and scored. In *Proceedings of the 14th International Conference on Mining Software Repositories*, MSR '17, page 511–514. IEEE Press, 2017. ISBN 9781538615447. doi: 10.1109/MSR.2017.45. URL <https://doi.org/10.1109/MSR.2017.45>.
- [2] R. Berrios, P. Totterdell, and S. Kellett. Eliciting mixed emotions: A meta-analysis comparing models, types and measures. *Frontiers in Psychology*, 6, 04 2015. doi: 10.3389/fpsyg.2015.00428.
- [3] E. Breck and C. Cardie. Opinion mining and sentiment analysis. In R. Mitkov, editor, *The Oxford Handbook of Computational Linguistics*, chapter 39. Oxford University Press, Oxford, 2 edition, 2017.
- [4] D. A. Fields, K. Pantic, and Y. B. Kafai. “i have a tutorial for this”: The language of online peer support in the scratch programming community. In *Proceedings of the 14th International Conference on Interaction Design and Children*, IDC '15, page 229–238, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450335904. doi: 10.1145/2771839.2771863. URL <https://doi.org/10.1145/2771839.2771863>.
- [5] E. Guzman, D. Azócar, and Y. Li. Sentiment analysis of commit comments in github: An empirical study. In *Proceedings of the 11th Working Conference on Mining Software Repositories*, MSR 2014, page 352–355, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450328630. doi: 10.1145/2597073.2597118. URL <https://doi.org/10.1145/2597073.2597118>.
- [6] M. S. Hajmohammadi, R. Ibrahim, and Z. Ali Othman. Opinion mining and sentiment analysis: A survey. *INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY*, 2(3):171–178, Jun. 2012. doi: 10.24297/ijct.v2i3c.2717. URL <https://rajpub.com/index.php/ijct/article/view/2717>.
- [7] B. M. Hill and A. Monroy-Hernández. A longitudinal dataset of five years of public activity in the scratch online community. *Scientific data*, 4(1):1–14, 2017. doi: 10.1038/sdata.2017.2. URL <https://doi.org/10.1038/sdata.2017.2>.
- [8] B. Liu. *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers, 2012. ISBN 1608458849. doi: 10.2200/S00416ED1V01Y201204HLT016.
- [9] A. Pak and P. Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of LREC*, volume 10, 01 2010.
- [10] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1–2):1–135, 2008. ISSN 1554-0669. doi: 10.1561/1500000011. URL <http://dx.doi.org/10.1561/1500000011>.
- [11] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, and Y. Kafai. Scratch: Programming for all. *Commun. ACM*, 52(11):60–67, Nov. 2009. ISSN 0001-0782. doi: 10.1145/1592761.1592779. URL <https://doi.org/10.1145/1592761.1592779>.

- [12] D. Rishi. Affective sentiment and emotional analysis of pull request comments on github. Master's thesis, University of Waterloo, 2017. URL <http://hdl.handle.net/10012/12728>.
- [13] S. Rosenthal, N. Farra, and P. Nakov. SemEval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518, Vancouver, Canada, Aug. 2017. Association for Computational Linguistics. doi: 10.18653/v1/S17-2088. URL <https://www.aclweb.org/anthology/S17-2088>.
- [14] S. A. Salloum, M. Al-emran, A. A. Monem, and K. Shaalan. A Survey of Text Mining in Social Media : Facebook and Twitter Perspectives. *Advances in Science, Technology and Engineering Systems Journal*, 2(1):127–133, 2017. ISSN 2415-6698. URL https://www.astesj.com/publications/ASTESJ_020115.pdf.
- [15] M. Thelwall. Heart and soul: Sentiment strength detection in the social web with sentistrength. *Cyberemotions*, pages 1–14, 01 2013.
- [16] M. Thelwall, K. Buckley, and G. Paltoglou. Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology*, 63(1):163–173, 2012. doi: 10.1002/asi.21662. URL <https://www.onlinelibrary.wiley.com/doi/abs/10.1002/asi.21662>.
- [17] M. Thoma. The Will benchmark dataset for written language identification. *arXiv preprint arXiv:1801.07779*, page 12, 01 2018.
- [18] A. Zeevaarders. An exploratory study of the learning progression of scratch users. Master's thesis, Open Universiteit, June 2020.

A

project_id	comments and replies
284032104	13967
318474570	11363
276660763	10348
318927435	8274
292793937	7871

Table 9: Top 5 of projects with most comments and replies

project_id	loves	project_id	remixes	project_id	favorites
305062819	13076	304292119	3023	292793937	11157
292793937	12527	302976767	1872	305062819	11011
302976767	12463	276660763	1753	302976767	9903
332477974	11467	292728003	1496	332477974	9332
285675327	10042	311402913	1492	318927435	9121

Table 10: Top 5 of projects with most loves, remixes and favorites

comment_id	replies
117642522	1788
117075320	912
118782760	809
116190007	789
114891556	665

Table 11: Top 5 comments with most replies.

Comment text	Frequency	Reply text	Frequency
cool	5662	Thanks!	7798
lol	3625	XD	3975
nice	2852	:D	3871
wow	1847	:)	3481
Cool!	1809	ok	3403
XD	1678	lol	3266
Nice!	1451	Thank you!	2644
hi	1421	yes	2191
cool!	1362	same	1840
LOL	1226	thanks!	1708

Table 12: Top 10 most common reactions for comments (left) and replies (right).

comment_id	comment_content
112776642	Omi.... *sobbbbs* I didn't knowyou get on to check messages ;w; it's good to hear some words from you, even if you don't get on often, or at all but....this is beautiful and I've always looked up to you as a really imaginative, beautiful, loving artist whose personality is just as gold as her creations... love you, with all my heart. Until I wander on your page again,,,lol... love you.
110624909	I just wanna know... some people have made games based on the book series Warriors (you should read it if you don't already) made with lists and variables. The lists and variables overlap, and the text bubbles are covered. We all really loved these games, and seeing them get ruined by the new update is really sad... maybe add a button where we can toggle text bubbles overlapping things?
109634863	'Scratch 3.0 is awful' :(:(:(Scratch 3.0 is NOT awful!!! Nice project, ScratchCat!

Table 13: All comments with both positive=5 and negative=-5. HTML entities are decoded.

comment_id	comment_content
105926616	REALLY FUNNY :P :)
114523152	SO ABSOLUTELY AWESOME!!!! KEEP MAKING THIS SERIES PLS!!!!
113440755	How about Itty Bitty for a name. :) your kitty is super adorable!! AAAaaaahhhh
112764068	That BenBen lockscreen is adorable?!?! *Socks faints in distance*
118294883	wow, loving these!! you're so talented with ink pens :,0

Table 14: A selection of comments with positive=5 and negative=-1. HTML entities are decoded.

comment_id	comment_content
111387443	TEARS:(
111404512	i was really scared by the thumbnail and then i clicked the green flag and my cardiac arrest was given bail
110129826	Were I not without online, you'd have a very angry Plant comin' to your doorstep, ready to rumble
129359336	This is actually very scary
110022502	ANIMAL ABUSE!!!!
119382038	Im sad :(where r u

Table 15: A selection of comments with positive=1 and negative=-5. HTML entities are decoded.

comment_id	comment_content
105628705	Aa hAPpY SCRAAtCH 2.0 BYe BYE dAY
105639351	U srsly cried?
105656614	Scratch 3.0 is wurd
105669811	I'll never let go SCRATCH 2.0,I'll never let go
105697498	Oh that's hot, that's hot.
105790209	i'll maek seezin thre an w bucks
105811901	THE SARCASM THO
105813626	oh never mind sorry
105903503	Amazing, i cried.
105939802	Howdy! im Flowey! Flowey the Flowers! :3
105965734	youve gotten better, i can tell

Table 16: A selection of comments that were identified as Scottish. HTML entities are decoded.

comment_id	original comment_content	decoded content
114854688	choc cmito doggo :>	choc cmito doggo :>
125281684	Fantastic!/Fant´stico!	Fantastic!/Fantàstico!
129609264	dogwood <3	dogwood <3
111205262	é o ani aniwoki aí	è o ani aniwoki à
115176821	wo ho i'm peter parker	wo ho i'm peter parker
108379537	"Luke Skywalker dislikes memes."	"Luke Skywalker dislikes memes"

Table 17: A selection of comments that contain HTML entities with the decoded text.