

Project Documentation
Hidden Knowledge of Stammbücher II
HKStam Summer Term 2023

Jakob Albrecht
Matrikel-Nr.: 124476

Jenny Döring
Matrikel-Nr.: 119611

Anastasia Koslova
Matrikel-Nr.: 125115

Jacob Lammert
Matrikel-Nr.: 121796

Ludwig Lorenz
Matrikel-Nr.: 120252

Bauhaus-University Weimar

September 2023

Content

1	Introduction	3
2	Project Description	3
3	Data Analysis	3
3.1	Data Sources	3
3.2	Analysis	5
3.3	Improving	6
3.4	Correcting the time data	8
3.5	Social Groups & Job Analysis	9
3.6	Image Analysis	9
3.7	Statistical Tests	10
3.7.1	Distribution Analysis	10
3.7.2	Precision Analysis	12
4	User Study & Mockup	14
5	Prototype	15
5.1	Database Search	16
5.2	Graph	17
5.3	Time Diagram	21
5.4	Language Diagram	28
6	Heuristic Evaluation / Expert Review	30
6.1	Search	30
6.2	Graph	30
6.3	Time Diagram	31
6.4	User Study Conclusion	32
7	Challenges & Solutions	32
8	Conclusion	33
9	Future Work	33
10	Bibliography	34
11	Appendix	35

1 Introduction

In our current digital era, interpersonal communication has evolved, rendering it effortless to remain in contact with friends via instant messaging platforms such as WhatsApp and by following their activities on social media channels like Facebook and Instagram.

However, what if we were to rewind the clock to a time before instant messaging and social media? Imagine a world where sending a message meant waiting days or even weeks for it to reach its intended recipient. In such a scenario, a different means of preserving and nurturing friendships would be necessary. Enter the concept of the “friendship album”.

Notably, the “Herzogin Anna Amalia Bibliothek” (HAAB) in Weimar is home to an extensive collection of friendship albums, representing the world’s largest compilation of such artifacts. Significantly, these friendship albums are digitally accessible via the HAAB website.

Building upon the groundwork established by the preceding “HKStam - Hidden Knowledge of Stammbücher” project in the winter semester of 2022/2023, which examined friendship albums as social networks and analyzed their distribution across Germany and globally, our current project seeks to advance this exploration.

This project focuses on analysing and improving the information, focusing further on exploring the concept of friendship albums as social network while analysing the time frame of the entries and more, like the language in which they are written.

2 Project Description

Friendship album entries usually contain popular citations or wisdoms the entry author wanted to share with the album owner. That catches one’s eye right away when looking into a friendship album. Despite that, such an album also contains hidden information, which may not directly get one’s attention, but probably this hidden information are what matter most. With information about the different authors, which entered in friendship albums, complex social networks can be revealed. If the locations specified below or next to the sign of the authors are analyzed, travel activities and interests of many different persons can be explored. If the dates of the entries are investigated, statistics regarding the time can be created. Questions like the following can be answered: In which time of the year or place in Europe the most entries were made? Does the distribution have something to do with a specific historical event? Which person interacted the most and who had the most contacts? Are these persons inside the same social group or do they differ?

These kinds of questions are all possible directions that could be explored during this project. One of the largest archive collection in the world with about 3000 albums is the “Herzogin Anna Amalia Bibliothek” (HAAB) in Weimar. About half of the holdings have been digitized and indexed in detail in the electronic library catalogue. One data set was provided to us for further investigation and analysis, because the goal of this project is the uncovering of hidden information inside the friendship albums.

3 Data Analysis

3.1 Data Sources

There are two different types of data sources used in this project. One one hand the friendship album entries and on the other the “Gemeinsame Normdatei” (GND).

The first and most important one are the 33893 friendship album entries given to us by the “Herzogin Anna Amalia Bibliothek” (HAAB) spanning from 1500 to 2022.

PPN	EPN	Materialart	Text	Blatt	GBV	Sprache	Jahr/Eintrag	Eintrag	Verfasser	Einträger	Beiträger	Seite	Kommentar	ExKommenta	ExKommenta	Enthalten in	Abrufcode		
1.029E+09	1.792E+09	Hsu	Texttxt, un	Blattnb	GBV102850f	ger	1570	[1570-1	[Stamm	buch	Schwan von	1028508123	Schwan von f	58	[1557-160	Illustration (Gouache), Wappen: in Gold	Enthalten in	stamm	
1.029E+09	1.792E+09	Hsu	Texttxt, un	Blattnb	GBV102850f	lat	1570	[1570-1	[Stamm	buch	Gutterer, Jac	1028507224	Gutterer, Jac	54	[1557-160	1], Bl. 67v.1	Enthalten in	stamm	
1.028E+09	1.792E+09	Hsu	Texttxt, un	Blattnb	GBV102847f	lat	1570	[1570-1	[Stamm	buch	Heininger, M	1028478968	Heininger, M	33	[1557-160	Illustration (Gouache), Wappen: in Gold	Enthalten in	stamm	
1.028E+09	1.792E+09	Hsu	Texttxt, un	Blattnb	GBV102841f	lat	1570	[1570-1	[Stamm	buch	Gremlich, Ulr	102841059X	Gremlich, Ulr	21	[1557-160	1], Bl. 23	Enthalten in	stamm	
1.028E+09	1.792E+09	Hsu	Texttxt, un	Blattnb	GBV102840f	ger	1570	[1570-1	[Stamm	buch	Stedynck, Jo	1028409958	Stedynck, Jo	19	[1557-160	Illustration (Gouache), Wappen: geteilt	Enthalten in	stamm	
1.028E+09	1.792E+09	Hsu	Texttxt, un	Blattnb	GBV102840f	ger	1570	[1570-1	[Stamm	buch	Böck, Matthia	1028407122	Böck, Matthia	15	[1557-160	Illustration (Gouache), Wappen: gespal	Enthalten in	stamm	
1.028E+09	1.792E+09	Hsu	Texttxt, un	Blattnb	GBV102840f	lat	1570	[1570-1	[Stamm	buch	Schreckenfus	1028405537	Schreckenfus	12	[1557-160	Illustration (Gouache), Wappen: der Sch	Enthalten in	stamm	
1.028E+09	1.792E+09	Hsu	Texttxt, un	Blattnb	GBV102839f	ger	1570	[1570-1	[Stamm	buch	Spiegel zum	1028394748	Spiegel zum	7	[1557-160	1] Illustration (Gouache), Wappen: der Sp	Enthalten in	stamm	
1.028E+09	1.792E+09	Hsu	Texttxt, un	Blattnb	GBV102838f	lat, ger	1570	[1570]	[Stamm	buch	Schad, Mich	722066872	Schad, Mich	ae	1	[1557-160	1] Illustration (Gouache), Wappen: in Blau	Enthalten in	stamm
839973594	1.581E+09	Hsu			GBV839973f	lat	1570	[Stamm	buch	SPMaximilian	07937400X	83997096X	S	21	[1515-161	Abschrift einer Eintragung für den un	in	683179754	
730913090	1.344E+09	Hsu			GBV730913f	lat	1570	[Stamm	buch	Winter, Joha	148931006	Winter, Joha	n	119	[1570-1579]	S. 119		in	68317792

Figure 1: An Overview on the structure of the friendship album entries

As illustrated in Figure 1, the given data consists of:

- a PPN
- a EPN
- the Material of the friendship album page
- the type of entry - text or picture
- the page of the entry
- one GBV (“Gemeinsamer Bibliotheksverbund”) number
- the used languages
- the estimated date when the entry was made
- the entry location with author and Date
- the author of the entry
- the person who wrote themselves into the friendship album
- possible co-authors of the entry
- the page, on which the entry was made
- any commentary to the entry
- possible examples of commentaries
- another possible example of a commentary
- the friendship album the entry was made in
- the owner of the friendship album with a code to find it

The PPN, EPN, material type, TXT, GBV and commentaries are currently not important to analyze, as their use case is quite specific and not in the scope of this project. For most database users it is important to see who made the entry, in which social environment they are, in which time frame they made their entries and in whose friendship album they have entered.

To achieve this, the second set of data, the “Gemeinsame Normdatei” (GND) is used. It includes large amounts of information of many historically important personalities. With the GND-IDs these people are linked to the friendship album entries from HAAB so no manual work to link the two sets of data together is required.

The second set of data, the GND, is in the entity_facts_db.json and consists of a context with a link to culturegraph.org with an explanation to the structure of each JSON file. The ID links to an entry from the “Deutsche Nationalbibliothek” (dnb) from the person with the ID. The DND also contains more information about the person like the name, different name variations, gender, date of birth and date of death, place of birth, place of activity, related persons, links to other databases with ID’s of the same person and most importantly the persons jobs and other biographical or historical information about this person.

Unfortunately many of the GND-ID’s are missing information or have irrelevant information as many ID’s don’t have a job or profession or have to detailed job description. For example one of the jobs for Helwig

Dietrich (115531777) is listed as “Hofarzt vieler Fürsten, u.a. in Hessen, Brandenburg u. Dänemark” which could have been shortened to court physician or court doctor or just simply doctor but this point will further be discussed in Section 3.3.

3.2 Analysis

When working with previously unknown data one has to know the quality of the data to perform informed decisions. Only when the quality of the data is known, steps can be taken to improve it if necessary and possible. This enables a better insight in the data as more of it can be used in the final analysis. As mentioned previously in Section 3.1 can the information in the friendship album entries be highly detailed. This however is not always the case as seen in Figure 2, where the data of all friendship album entries is sorted according to the amount of information contained. This sorting is first performed for the given name information, and then either according to the availability of the name or the included profession.

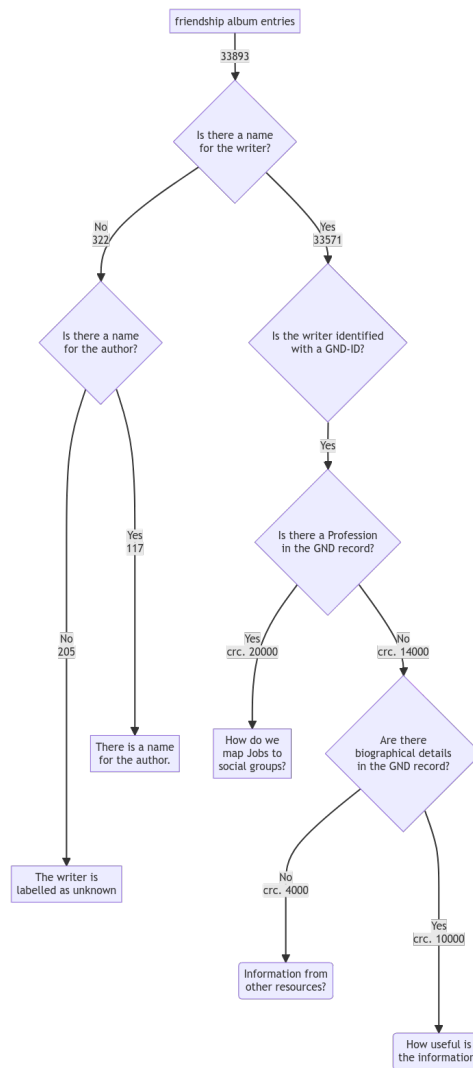


Figure 2: Decision Tree for the entities data based on their information wealth.

Analyzing the data in such a large grained manner allows for a large scale quality evaluation of the friendship album entries. Improving the data is best done where the impact will be maximized. For the unknown authors for example only 322 out of 33893 entries in total do not have a name. Adding names would, in most if not all cases, have to be done manually with access to the physical friendship albums. One branch in the tree is perfect for improving the friendship album entries. Using the biographical data

of around 10000 entries, professions could be added to these authors who previously had no information on their profession. Assuming the professions in the biographical data is detailed enough this process can also be automated as mentioned in Section 3.3.

The profession data of around 20.000 to around 30.000 entries if improved can then be used for analyzing the social groups as explained in Section 3.5.

3.3 Improving

Improving a set of data can usually be done either manually or automatically. Manually improving the data would require not only access to additional knowledge but more importantly also time and human resources. Automatically enhancing the data however can be done comparatively cheap, easy and fast. One major disadvantage is the expected error which can usually be reduced with growing complexity of the program. It may be necessary to check multiple different data sources for inconsistency in order to provide acceptable results. Grammar and context can also be difficult to interpret without additional knowledge about the time period, customs or location.

In the friendship album entries there are 33893 entries in total from which around 20000 have an assigned profession and 14000 do not. From the 14000 entries without an assigned profession around 9000 have biographical details in the GND record.

In order to answer different questions about the social groups in the dataset it is advantageous to know the professions of as many people as possible. Using the biographical information one could try to assign a profession to a given person. This would now take a large team, or a long time to do manually as there are around 9000 entries to process.

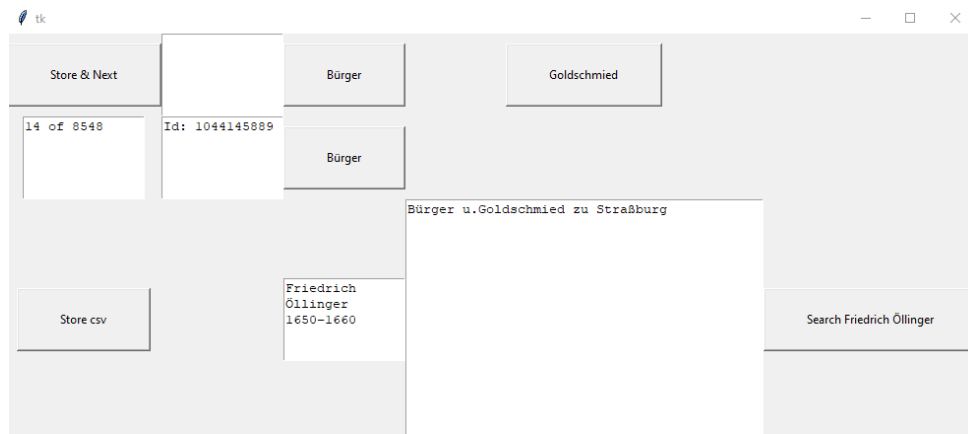


Figure 3: Python based tool for entity data processing. A profession can be assigned based on the biographical data.

To accelerate this process, a tool was built in python which aims at accelerating this assignment. Using the `tkinter` library it generates buttons and text fields for displaying information and enables additional interaction capabilities. Figure 3 displays a screenshot of the tool which shows the data of an entry with the GND-ID 1044145889.

On the top left there is a button to save the changed profession and to go to the next entry. The current progress as well as the GND-ID and the name of the person is also visible. A large text field on the bottom right includes the historical and biographical information. This information is mainly used to assign a profession to the given person. To its right there is a button to quickly perform a search for the person on google. On the bottom left a button “Store csv” is visible which stores the currently assigned professions for each entry in a csv file.

The most important buttons are perhaps the automatically generated buttons in the first and second row above the biographical information. They include suggestions for professions based on the biographical data. The first row includes suggestions using professions of the 20000 entries with a professions. The second row “remembers” user generated professions.

The inner works of this tool are relatively simple. First a list of all possible professions is generated based

on all professions in the 20000 entries with a profession. Secondly, this list is used to generate suggestion for assignable professions based on the biographical data for each entry without an assigned profession. This can be seen in Figure 3 where “Bürger” and “Goldschmied” are professions which were found in the biographical information of Friedrich Ölinger. They are both in the first row as the professions “Bürger” and “Goldschmied” were previously assigned as professions for old entries and then stored in the database. The suggestions in the second row compare all manually added professions during the runtime of the program to each word in the biographical information. If one previously assigned profession matches a word up to around 90%, the previously stored word will get added to this suggestion row as a button. The second row in Figure 3 containing only “Bürger” as a suggestion learned from a previously, manually entered suggestion that Bürger is a valid profession. When a button is clicked, the empty field to the right of “Store & Next” gets filled with the clicked profession.

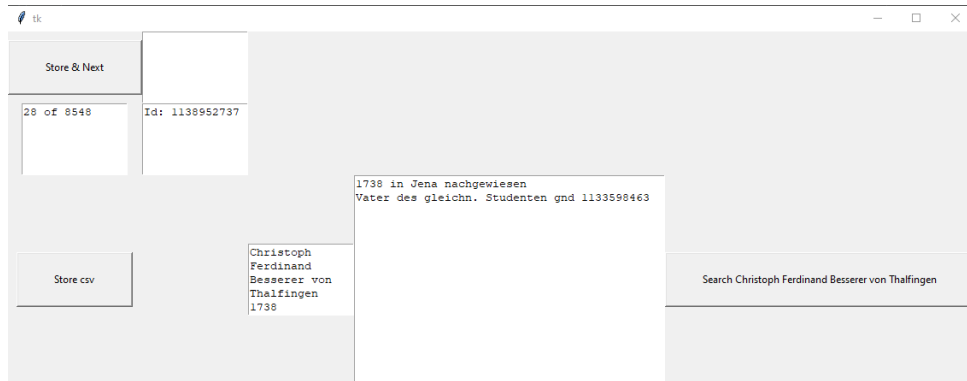


Figure 4: No profession is found as “Studenten” was never used before

If no profession is proposed, the entry can either have no useful biographical information, or there exists biographical information but the included profession has never been assigned before. If it has never been assigned before, the profession has to be added manually to the previously mentioned text field. After the profession is written in the empty text field and “Store & Next” is clicked, it is internally added to a list of manually added professions. Should the profession now appear again somewhere in the biographical information, the previously unknown profession now appears as a suggestion in the second row. If no useful information exist as illustrated in Figure 5, a google search can be performed if the name and optionally the birth or death date is known. The birth or death date is often needed as there may be many different people with the same name or who may even have lived during the same time. The result of this search is illustrated in Figure 6. The search may however not always be successful as shown in the example.

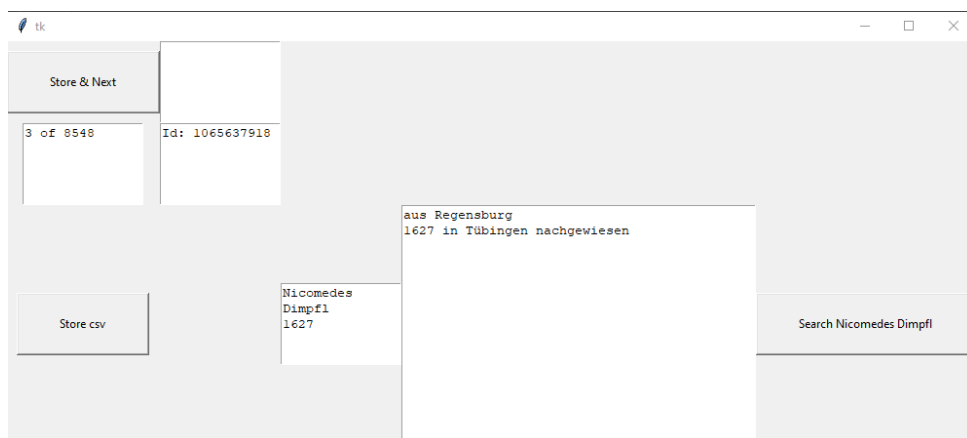


Figure 5: No useful biographical information is given. An additional google search with the Search button may be performed.

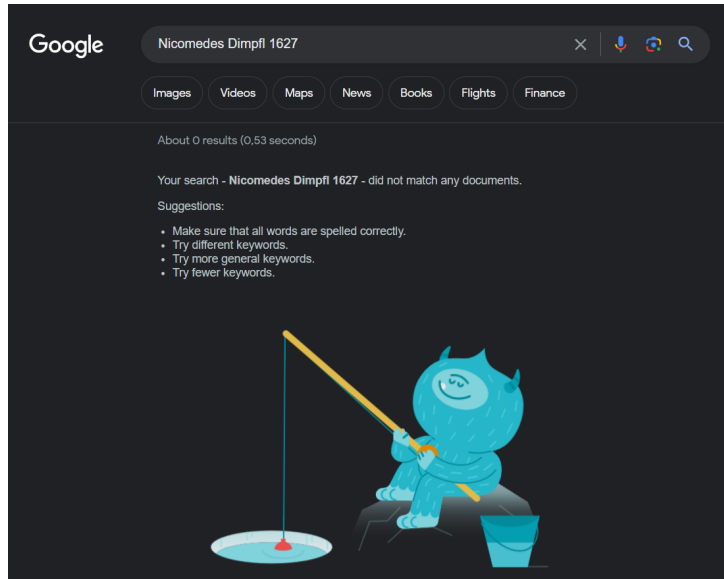


Figure 6: Search for Nicomedes Dimpfl with unproductive results.

The process of assigning professions to the 9000 entries was not performed in full, however a random sample of 100 entries was generated. From this sample the profession for 27 entries could be assigned automatically and for 47 it could be performed manually. The 27 automatically assigned professions varied in some cases from the manually assigned ones. When the automatic assignment is performed multiple different professions are accepted while the manual assignment only accepts one profession. When assigning a profession manually the process is often subjective. Accepting all possible professions may lead to a more objective result as a higher number of correct professions compared to incorrectly assigned professions may be assigned.

It can be extrapolated that around 5000 of the 9000 entries could get a manually assigned and around 2500 could get an automatically assigned profession with a confidence of around 95% .

One problem for the automatic assignment is as previously mentioned the missing knowledge for the German grammar. Figure 4 for example could lead to an automatic assigned profession of “Student” even though the profession in this case is unknown. The entry belongs to a Father with GND-ID 1138952737. He has a son with profession “Student” and GND-ID 1133598463.

A well understood context therefore is key to correctly assigning the correct profession. Understanding the context is a problem which may be solvable by machine learning approaches.

When assigning professions manually there are also some smaller issues cultural and societal issues. Exotic names for professions or words can appear which are no longer used. It may also be problematic to label the profession of a woman as “Ehefrau” or “Tochter” as they may not have had other mentioned professions.

3.4 Correcting the time data

Especially the time data was very inconsistent and needed to be standardized. There was the situation, that not all time information were given in a format like in example, `dd.mm.yyyy`, where information about the year, month and day as well are available. In many cases, additional digits or letters were used to express the time information. Also in a not negligible amount of data set entries the time information was given in Roman numerals and needed to be transferred into Arabic numerals.

This time data correction was very time-consuming, especially because of many individual cases, which could not be handled through a script. Nevertheless, for most of the entries correcting the time data was possible by using a script. It was decided to implement it in `Python`. The goal of the data correction was to get an uniform data format. In many cases, the day information was missing or only the year of the entry was given. Therefore it was necessary to transform the time information into the following formats sorted from highest (1) to lowest (5) priority:

1. dd.mm.yyy
2. mm.yyyy (if no day information was given)
3. yyyy (if neither day nor month information was given)
4. yyyy-yyyy (if only an year interval was given)
5. unknown date (if no time information was given)

To display all these kinds of information, for example in the monthly x-axis view (see Section 5.3 for more information), the following strategy had been performed:

An information with month and year or day, month and year specification increased the amount of entries for the corresponding month by one. If there was only a information about the year, the amount of each month was increased by $\frac{1}{12}$ to not loose this information. Unfortunately the time data entries in the formats of priority 4 and 5 could not be taken into account within the scope of this time diagram.

3.5 Social Groups & Job Analysis

In order to achieve a classification into social groups, the job description that every author possesses due to its GND-ID was first filtered out. After that and counting the amount a job description occurred, data was filtered again in order to obtain an overview of the job descriptions that were most common.

It was decided to only consider the job descriptions at first that occurred 50 or more times, presenting a set of 86 jobs. Afterwards, the job descriptions were sorted manually while assigning a suitable headline for each category and subcategory. Due to the jobs being in German, German is also used for the category names. The final categorisation can be found in the appendix.

One issue that occurred is that it was not certain whether the classification that was chosen is appropriate for that time when friendship albums were collected. A doctor for example could have had a different academic background and meaning than it has today. For those purposes, specialists in sociology need to be consulted in order to validate the categorisation of social groups which were chose.

Another issue occurring is that some authors had multiple job descriptions listed, and thus, it was difficult to decide where authors should be assigned to multiple social groups or just one.

3.6 Image Analysis

While Section 3.2 focused on already digitally available data and its quality, this section will take a short look at the underlying physically available data.

Transcribing a friendship album has traditionally been a very precise attentive task performed by highly skilled individuals sometimes working one letter at a time. This is a very time consuming process which, with the help of modern image analysis tools may soon be performed much quicker with less manual labour required.



Figure 7: Image illustrating reCaptcha with two different text inputs “Levelers” and “critics” [1].

Figure 7 illustrates an old reCaptcha which one may have already encountered on the internet. Captcha stands for (Completely Automated Public Turing test to tell Computers and Humans Apart) which is used to mitigate large automated bot attacks on many servers.

The reCaptcha is a revised version of the original Captcha which aims at distinguishing a real user from a bot, but also at using the work of the user in a productive manner. The two words in Figure 7 for example are used for automated text transcription. One word is a control word which is already known while the second image may contain a not yet known word which is scanned from an old document. If the control word is put in correctly, it is assumed that the second word is also inserted accordingly. This process is then repeated many thousand times a day and the resulting data is used to train machine learning models for OPR (optical character recognition). Google Books used this technology to scan around 40 million books worldwide [2].

Image analysis was first performed in python where the python scripts wrongly identified dark spots of the paper as letters. Therefore a more sophisticated method was needed which was provided by google. In a short test GoogleLens, a mobile version of a text scanner provided promising results as even Greek characters of an entry could be transcribed. Copying the text from the app for each scanned page however could not yet be automated. The time improvement compared to manual labor would however be immense. But scope of the implementation is too big for this project but the technology is promising as proven by googles book digitization program.

Fully digitizing the entries would need to be done in a follow up project with the collaboration of the HAAB.

3.7 Statistical Tests

3.7.1 Distribution Analysis

Researching statistical tests was important in order to prove the validity of the automatic algorithm developed in order to assign jobs automatically to social groups. Hence, this Section is devoted for the research outcome on statistical tests. The content for testing for a normal distribution [3] and parametric and non-parametric tests [4] are taken from the website datab.

Testing for a Normal Distribution

In order to conduct a statistical test, it is first crucial to test whether the data follows a normal distribution. If this is the case, then parametric tests will be used in order to test the correctness of the hypothesis. If not, non-parametric tests will be used.

Hence, it is necessary to test the null hypothesis that the data is normally distributed with a significance level of 5 percent. Three different tests are highlighted in the literature: the Kolmogorov-Smirnov Test, Shapiro-Wilk Test, and the Anderson-Darling Test.

Kolmogorov-Smirnov Test

The advantages of the Kolmogorov-Smirnov Test is that it is possible to identify the type of distribution without any specific parameters. It is easily applied to a set of data and delivers robust results. This test can be used to compare distributions as well. A disadvantage is that the test is sensitive in regards to standard deviations. The results are also not significant with small data sets.

Shapiro-Wilk Test

The advantages of the Shapiro-Wilk Test are that it is a good fit for finding out whether the data follows a normal distribution. The test can be applied to both small and big samples and produced reliable results that can be easily interpreted. One disadvantage is that the test becomes very sensitive for samples with more than 2000 data points in regards to standard deviations, meaning that the null hypothesis can be very easily rejected.

Anderson-Darling Test

The Anderson-Darling Test is able to identify multiple distributions, besides the normal distribution. Even small samples lead to reliable results. There are also multiple critical values available for the tests which means that the test can be easily adapted to a different significance level. A disadvantage is that the test is more complex than the Shapiro-Wilk Test, especially when comparing multiple samples. If the sample is too low, then the probability that the null hypothesis will be rejected is quite high. Additionally,

the test requires that the initial assumption of what kind of distribution the data follow is correct as an incorrect assumption leads to misleading results.

For a sample of 100 data points, all tests presented here are applicable but it is advised to be cautious as many authors in our data were not assigned a profession (approximately half of the authors). If it is believed that the sample will follow a normal distribution, then it is advised to use the Shapiro-Wilk Test. If not, then it would be best to use the Anderson-Darling Test.

Parametric Tests

If the data follows a normal distribution, parametric tests should be used to test the validity of a hypothesis. Possible test methods are the t-Test and an analysis of variances (ANOVA).

T-Test

The t-Test is conducted on one sample and tests whether the sample mean proves to be significant, meaning how much it differs from a known mean. There is also the possibility of conducting a t-Test on two samples and compare the two sample means. An advantage is that the t-Test can be used to identify whether two groups differ significantly but a disadvantage is that t-Tests are usually used for research with a control group such as testing a medication and it is not common to apply this test to testing the validity of algorithms.

Analysis of Variances (ANOVA)

An analysis of variances checks whether there is a significant difference between more than two groups. An advantage is that the test intends to allow using multiple measurements (multiple samples), and its aim is to compare all sample means with one another. A disadvantage is that the test is intended for continuous variables such as income (for example testing the income in specific living areas) and would hence not be suitable to test the validity of an algorithm.

Non-Parametric Tests

If the data does not follow a normal distribution, non-parametric tests should be used to test the validity of a hypothesis. Possible test methods are the Mann-Whitney U Test and the Wilcoxon Test.

Mann-Whitney U Test

This test follows the same aim as the t-Test but is specifically designed for data that does not follow a normal distribution. Thus, it has the same advantages and disadvantages as the t-test. The only importance is that the samples are independent from one another.

Wilcoxon Test

The Wilcoxon Test analyses whether two dependent samples significantly differ from one another. It is important that the samples are dependent, and the advantage is that the tests intend to have multiple measurements.

Overall, it was difficult to find a test that was able to validate the algorithm that was developed. After researching, it was uncertain to conclude what the right method would be as it was difficult to find an example from the past where a similar type of algorithm was tested.

The greatest difficulty was to name the exact variables as a continuous variable was needed where it was possible to find a mean from the data. But the result was only stating whether the algorithm made a correct or incorrect assignment. Further research is encouraged for the future, especially in the area of binomial distributions.

3.7.2 Precision Analysis

The previous section illustrated that different types of statistical tests could not be applied to prove assumptions over the underlying distribution type. Opposed to that statistical methods can be effectively employed to draw conclusions about simple statistical features like the accuracy of machine labelled records within different social groups. In classification problems the precision and recall rate are key performance indicators to quantify accuracy. The precision rate measures the fraction of relevant retrieved elements among all retrieved elements. The recall rate measures the fraction of retrieved relevant elements among all relevant elements.

In the context of this report, for a given social group, denoted as s , the precision rate signifies the proportion of correctly labeled individuals with social group s among all individuals labeled as such. Analogously, the recall rate quantifies the proportion of correctly labeled individuals with social group s among all individuals who actually belong to that social group. It is important to note that calculating the true recall rate requires manual annotation of the entire dataset to determine each person's membership in social group s . In contrast, for calculating the true precision rate, manual annotation is only required for the smaller subset of individuals already labeled as belonging to social group s by the machine.

In both cases, manual annotation is not feasible given the dataset's size and the project's scope. Instead, a statistical hypothesis test can be developed to evaluate assertions regarding the machine-assisted social group labeling. Hypothesis testing is a statistical procedure that enables researchers to assess the credibility of a study by examining an assumption related to a population parameter. In this context it is tested if the assumed probability that a person labelled with a certain social group is actually not supposed to be assigned to that group. This probability corresponds to the concept of the precision rate. A hypothesis test to assure that the machine-assisted labeling of the social group "Weltliches Akademisches Umfeld" is precise enough is presented here as an example.

Hypotheses: Because each labeled data record is considered a binary outcome, such as "correctly labeled" (success) or "incorrectly labeled" (failure), a binomial hypothesis test is conducted. A binomial hypothesis test typically follows a structured process. Firstly, it involves defining two hypotheses: the null hypothesis (H_0) and the alternative hypothesis (H_1). H_0 represents the bad case that our annotation does not hold up well compared to manual annotation and has worse precision. A choice must be made on how "well" should be quantified in the hypothesis.

H_0 : Individuals in the social group "Weltliches Akademisches Umfeld" are actually part of this social group only in a maximum of 95% of the cases. ($p \leq 0.95$)

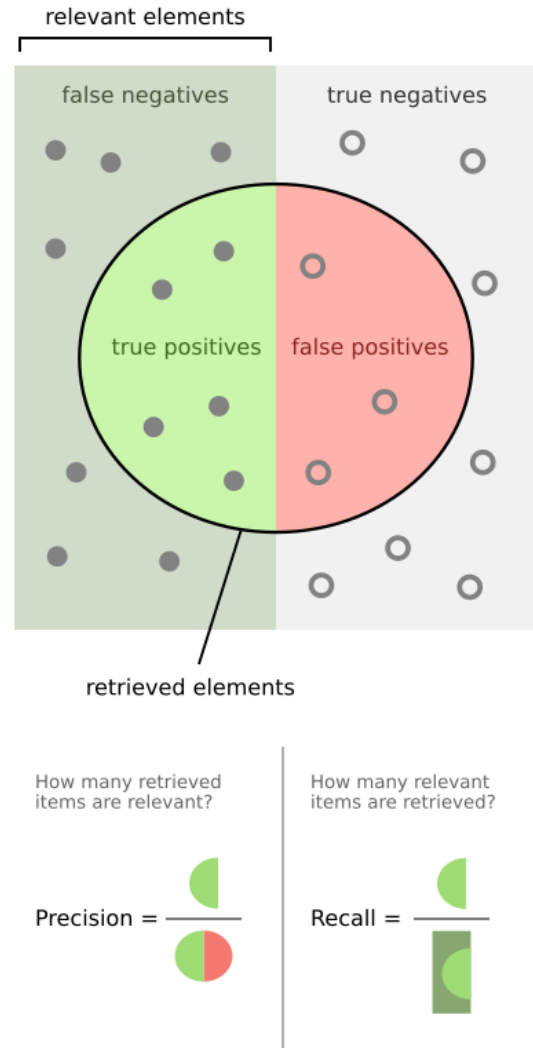


Figure 8: Diagram representation of Precision and Recall. Walber, CC BY-SA 4.0, via Wikimedia Commons

Why start with the bad news? Because the test aims to find such strong evidence against H_0 that the null hypothesis can be disregarded and the good case, the alternative hypothesis H_1 , be accepted instead.

H_1 : Individuals in the social group “Weltliches Akademisches Umfeld” are actually part of this social group in more than 95% of the cases. ($p > 0.95$)

The alternative hypothesis covers the right side of the probability spectrum, hence this type of test is referred to as a right-sided test. This means that the primary interest lies in detecting if the precision rate is greater than our assumed probability of false positives. If the statistical analysis provides strong evidence supporting the alternative hypothesis, it suggests that the precision rate is indeed higher than what was initially assumed. On the contrary, if it fails to find such evidence, it might be concluded that the precision rate is not good enough.

Errors: However, it’s essential to note that even though these conclusions are supported by statistical evidence, there remains a small probability that they are incorrect. This is because no statistical test can guarantee absolute certainty. There are two types of errors that must be considered: The first error type occurs when the test decides for the alternative hypothesis even though the null hypothesis is true in reality. (Type I error) This is the most fatal error type for this investigation since it aims to be as sure as possible when deciding that the given annotation is precise more than 95% and therefore well suited for further research.

The second error type occurs when the test favors the null hypothesis even though the alternative hypothesis is true. (Type II error)

	H_0 is true	H_1 is true
Test favors H_0	Correct.	Type II Error β
Test favors H_1	Type I Error α	Correct.

Significance level: A significance level at $\alpha = 0.05$ is chosen to set a boundary for the Type I error. This implies that if the null hypothesis is, in fact, true, this test will make an incorrect decision (rejecting the null hypothesis) only 5% of the time. In other words, there is a 5% chance of committing a Type I error by concluding in favor of the alternative hypothesis when it’s not actually true.

Conversely, if the reality is such that fewer individuals are correctly assigned to the social group than our assumption suggests, this discrepancy can be detected with a 95% probability. This results in the test being quite sensitive to deviations from the assumed precision rate, allowing to identify such deviations with a high degree of confidence.

It is important to note that this 95% probability of preventing a Type I error is different to the 95% probability mentioned in the hypothesis itself.

Sample: Next, a sample of 25 labeled individuals from the social group “Weltliches Akademisches Umfeld” is collected, where each labeled individual again is considered a binary outcome, such as “correctly labeled” (success) or “incorrectly labeled” (failure). With manually annotating the sample it is concluded that 22 individuals were labelled correctly but 3 individuals do not belong to the social group.

Test: Based on those results a binomial test is performed with help of the `scipy` library in python.

Example Code:

```
from scipy.stats import binomtest

result = binomtest(22,25,p=0.95, alternative="greater")

result.pvalue
>>> 0.9659
```

The calculated p -value is with 96.6% way above the tolerated α -level of 5%. From here it can be tried to fool the test by drawing a sample repeatedly in hope of achieving one sample that satisfies our hypothesis.

However, even if all individuals in the sample of size 25 are correctly assigned to the social group, the p -value will result in 27.7% which is still out reach for the chosen tolerance zone.

Instead, can be tried to increase our sample size until significant evidence for the alternative hypothesis is actually possible. However, for high probabilities like “more than 95%” that is demanded in the given alternative hypothesis the required sample size can grow very quickly. This leaves the question if the probability demanded in the hypothesis was even feasible to test in the first place. For the assessed sample the confidence interval contains the highest possible probability for the hypothesis that still does not violate the chosen α -level.

Example Code:

```
from scipy.stats import binomtest

result = binomtest(22,25,p=0.95, alternative="greater")

result.proportion_ci()
>>> ConfidenceInterval(low=0.7182774920222817, high=1.0)
```

The `proportion_ci()` function provided by `scipy` is used to calculate the confidence interval. The confidence interval suggest that it can be claimed with significant evidence that individuals in the social group “Weltliches Akademisches Umfeld” are actually part of this social group in more than 71% of the cases. ($p > 0.71$)

Conclusion: The hypothesis test was conducted to estimate the precision rate of machine-assisted social group labeling. In a sample of 25 individuals, 22 were labeled correctly by the machine-assisted approach. The statistical analysis, with a significance level of $= 0.05$, did not provided strong evidence to support the alternative hypothesis, that precision is higher than 95%. Instead a confidence interval indicated that individuals in the social group “Weltliches Akademisches Umfeld” are actually part of this group in only more than 71% of the cases. **This effectively weakens the informative quality of any other social group based research on the dataset.** It is recommended to review the social group annotation of the dataset again before conducting further research.

While these findings provide valuable insights into the accuracy of the labeling process, it’s important to acknowledge the inherent uncertainty in statistical testing. Despite the high degree of confidence in the drawn conclusions, there remains a small probability of error due to the chosen significance level. Nevertheless, these results offer a solid foundation for understanding and improving the precision of machine-assisted social group labeling in this project as well as providing a framework for future annotation evaluation.

4 User Study & Mockup

Before the implementation of the tool prototype could have been started, the user groups, which could use the tool, and of course the goals and functionalities of the tool needed to be clarified. Another main aspect necessary to consider beforehand was the user design of the future tool.

Who wants to explore entries of friendship albums or the albums itself? That question was the base of the investigation of the potential user groups of the tool.

Some people may be interested in genealogy. Therefore, they could want to search for specific persons, i.e. of their family trees within the tool. Hence, this people were grouped as *interested laymen*. Further on, there may be *academics*, which, for example, deal with friendship albums within their study program and want to investigate more intensively when and where entries were made. They could want to explore the time spans, in which persons entered in friendship albums or the geographic spaces, where many enterings could be located. Furthermore, different people could want to visit different exhibitions of friendship albums. In some kind of the future it may be possible, that the developed tool is presented as QR code in there. *Visitors*, that scan this code probably do not have academic background in this research field and are searching for simple and clear context, which complements the exhibition. Despite

that, of course it can happen, that people, who did not want to use the tool (*lending audience*), find the way to it during a web search. This persons therefore need an easy way out of this tool.

With these considerations in mind, the application goals were addressed. For the *interested laymen* there should be available the possibility to search specifically for individual persons. Therefore, a search bar, that enables to search by entering a name or a GND-ID, was necessary. Also the name of one person in an alternative spelling could be helpful to find an individual person. Life data and the location of birth or other personal data also are good properties to determine, if the person found is the searched person by the user. Also some relations to other persons could be helpful to identify an individual person. Inside a graph relationships of persons, who entered inside a friendship album of another person could be nicely displayed as nodes and edges.

For the *academics* as well as for the exhibition *visitors* a presentation of overall knowledge could be useful. Thus, general diagrams, e.g of the time data or the languages used inside the friendship album entries were needed. Time intervals, for example, as well as a filter for the social groups should be layable like a mask over the graph and the other visualizations. Furthermore, the visualizations should be easy to understand overall.

For all user groups and especially the *lending audience* also an easy handling of the tool is mandatory. Hence, a multilingual website and easy control, for finding for example the button back to the “Klassik-Stiftung”, would be convenient. Another part could be a FAQ, where the users can read, what the purpose of this tool or website is. Digital sources could be linked or, i.e. the digital versions of the friendship albums could also be displayed of the page. Also the possibility to use the tool within different kinds of hardware or operating system would be very user-friendly.

Combining this multiple ideas led to a first paper prototype, which should try to cover all functionalities, that were considered as important. The paper prototype consisted of an overall view over the whole data set on the first page and another page with filtered data. The filtered data thereby could be a specific persons or friendship album as well as time intervals, years or a social group. Especially for the filtered data many interactions between the different diagrams should be possible.

5 Prototype

After the analysis of the potential user groups (see Section 4) the decision of implementing a prototype with different diagrams, that express different aspects of the friendship album data set, was made. Because the analysis and correction of the raw data set (see Section 3.4) was implemented in **Python**, the implementation of the prototype was done in **Python** as well. Different kinds of the data, for example the language or time distribution of the entries as well as a distribution of the different social groups, were tried to be implemented.

Because there should be developed a kind of tool, which can be explored interactively by a potential user, this approach was revised once again. It was decided to switch the programming language to **JavaScript**. With **JavaScript** and furthermore the help of **Github Pages** it was possible to develop a web service, which can be used independently from the users hardware or operating system.

To set up a web application the package manager **npm** and the build tool **snowpack** were used. Moreover, library to enable implementing diagrams easily, but with strong results, was essential. Hence, **d3** was chosen for this purpose.

After every push to the repository a Github action script is triggered that compiles the web application online on a virtual linux server via **npm build**. The result of the web application is then pushed on Github pages where it can be viewed publicly.

Because of time issues it was not possible to implement all diagrams and functionalities, which were brainstormed at the beginning of this project. In the following subsections the parts of the final prototype are presented.

5.1 Database Search

The database search enables the user to search for a person using their name or GND-ID. After a successful search, specific information about the individual is displayed namely the name, GND-ID, jobs, birth year, death year, the amount of entries, and the number of books owned. Furthermore, when searching for a person, the search is supposed to also impact all other graphs displayed on the page, enabling an interactive and dynamic web application.

First, it was crucial to filter the data in order to enable the search for individuals. Two data files were available: One was the CSV file “`stammbuchdaten_1500-2022_preprocessed`” that contained all the information about the collection of friendship albums of the Klassik Stiftung. The other was a JSON file called “`entity_facts_db`” that was created after filtering out the GND-IDs from the main CSV file and exporting the entity facts about each GND-ID.

In the JavaScript file “`filter_data_for_search`”, multiple functions to filter the data were implemented to gather the information relevant for the search. The goal of the code is to process and combine data from different sources to create a structured JSON object containing information about individuals, their jobs, books they own, and entries they have authored. The code utilises various utility functions to filter, group, and transform the data into the desired format.

The code imports data from other JavaScript modules using ES6 module syntax. The data includes the variable “`data`” containing all information from the CSV file, the variable `data_entries` containing all information from the JSON, and the function “`filter_id_to_id`” producing a data set with entry author, book owner, and EPN-ID.

The function “`get_info`” filters information (name, GND-ID, jobs, birth and death years) from the JSON object, entailing the entity facts with the GND-IDs. Furthermore, the function “`store_stamm(data)`” filters and organises data by grouping books by their owners. It creates an array that inserts an GND-ID with an array entailing the owned book and adds the EPN-IDs to the array if the person owned multiple books.

The function “`filter_entries_per_id(filtered_id_to_id)`” takes as input the array with EPN-ID, owner and entry author and produces an array that contains the GND-ID and an array with all the books in which the person authored something.

The function “`who_no_job(stamm_data, name_job_data)`” was used to identify individuals who have no associated jobs or professions yet that were extracted from the entity facts. It was implemented because the book owners were first neglected when producing the JSON file with entity facts. Currently, the function has no use and will probably be not used in future implementations.

The code lines that are commented out in the end indicate the steps for data processing, which include calling the mentioned functions and performing data manipulation. The commented `console.log(final_data)` line would print the consolidated JSON data to the console. Afterwards, it can be copied and inserted in the JSON file “`search_data_per_id`”.

The code for the actual database search can be found in the `search.js` file mainly. It is a JavaScript program that interacts with an HTML document and uses the `D3.js` library to perform dynamic search and display operations.

The program imports the `D3.js` library and assigns an initial value to the variable “`gnd`”, later containing the author’s GND-ID that is searched for. Afterwards, the data from the JSON file is loaded to access the relevant information about all authors and book owners for the search. By implementing an event listener on a “`Search`” button, the function named “`search`” is triggered when clicked in order to start the actual search of the author within the database.

In the search bar, the user is able to provide input in an input field to search for an author. Both name and GND-ID can be used for the search. The search function is then able to filter the loaded data to find matching GND-IDs or names based on the user input. In order to narrow down the search, the program updates a data list with the options of matching names and GND-IDs.

Initially, several HTML elements for the search such a name, jobs, birth and death are hidden so that the user is able to completely concentrate on the search tool. After pressing the search button, those elements appear using the `.style.display` property, displaying the information specific to the author.

Furthermore, another function called “`searching_book`” is implemented in order to be able to select a friendship album that a person owned. If the individual did not own any books, then the option to select is hidden. If the person has at least one, it is possible to select a book for further information. At the moment, further functionality is not implemented.

The variable “`gnd`” is exported for the use beyond the search function. It will be used for the graphs that are part of the web application to dynamically update their data as well. The goal is that when the user searches for a person, the whole web application is dynamically update, showing the data and graphs relevant to the person.

Overall, the code uses `D3.js` to dynamically update the display based on user input, showing information related to matched names or GND-IDs from the loaded JSON data. It hides and reveals various HTML elements to display search results or an error message when no match is found. The program also provides a way to search for book information and exports the variable “`gnd`”.

5.2 Graph

A Graph is a set of vertices connected by a set of edges. Not every vertex has to have a connection to a different vertex. A vertex can also have multiple connected edges which depending on the use case may even have arrows indicating a direction. They are often employed for visualizing a network, operations, a control flow or a supply chain.

Friendship albums contain knowledge similar to a friends list in social networks such as facebook, twitter, myspace or schülerVZ which can be visualized using graphs. For dense graphs it is often a difficult problem to generate easily explorable graphs. In the case of friendship albums however the generated graph will not be too dense to visualize completely. The total number of digitized entries compared to the total number of physical entries is low enough to generate an explorable graph.

To generate a graph two different sets of data are needed. One containing all the possible authors of entries and the owners of the friendship album. And one which contains all entries in each book. The two sets of data can be linked using the GND-IDs which enable unambiguous identification for a person in both data sets. The data sets are then processed in JavaScript to generate a list of all entries from one person in the friendship album of another person. The library D3s computes from the list of vertices and edges a graph. These vertices and edges are the backbone of the visualization. Computing the graph takes a few seconds. First the vertices get a weight assigned. And Secondly the connections get simulated as springs in a dynamic system. The simulation converges to a static graph after a few hundred iterations. This has the advantage of squishing the higher dimensional graph onto the two dimensional screen while spacing the vertices out in an appropriate manner. For optimisation purposes these steps will only have to be computed once. When reloading the website, the graph including the vertex positions and edge connections is reloaded from the stored file on the hard drive.

After computing the graph it is also important to visualize it appropriately. Visualizations with JavaScript are usually done by appending graphical objects into the HTML code of a website. HTML describes which buttons and objects appear on which position and color on the screen. But as the number of vertices and edges increases with the graph size, this quickly leads to poor performance. Because up to many thousand vertices and even more edges have to be in present in the HTML document at all times, the website is slowed down drastically. In order to speed up this processing time, the vertices and edges are no longer stored in the HTML code of the website. To visualize the graph, a function is now needed which efficiently draws shapes such as lines or circles only when they are visible on a canvas. This gives us only have one additional element in our HTML code, the canvas the graph is drawn onto.

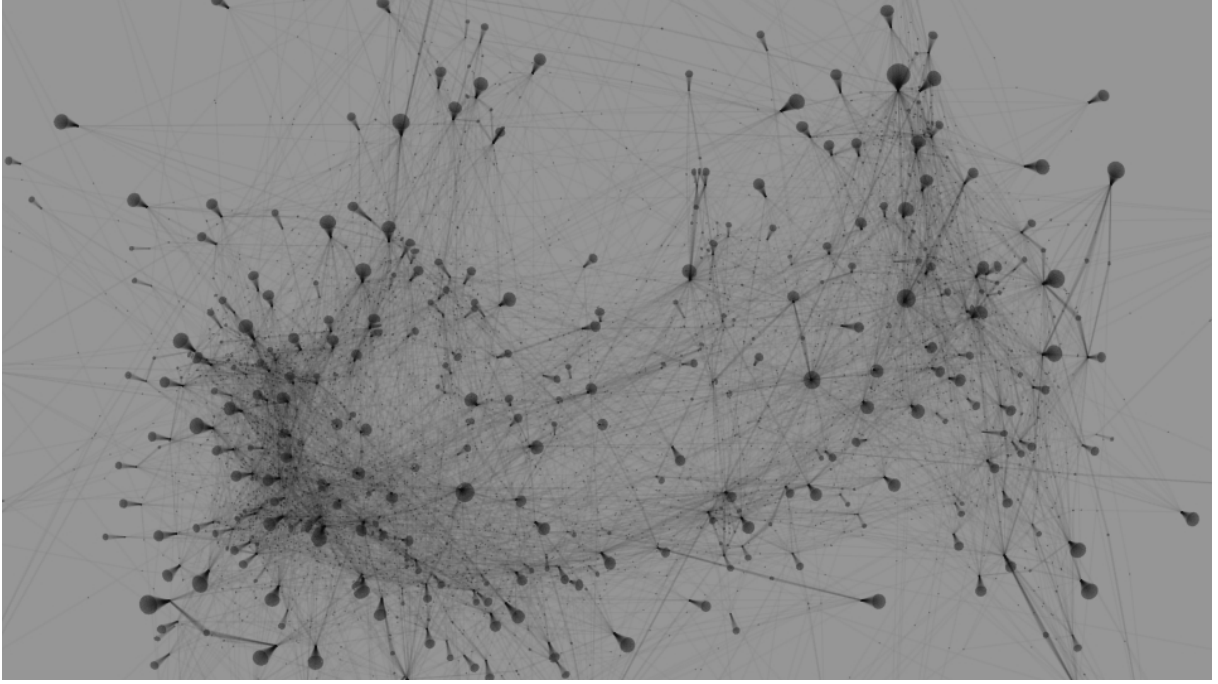


Figure 9: Friendship album network depicted as graph which is generated using JavaScript.

As the graph is large and interconnected as illustrated in Figure 9, tools for interactive navigation over the data are needed. The first tool is navigation which enables the user to zoom in, zoom out and move around the graph by a click of the left mouse button and by moving the mouse. This tool alone allows us to explore the vast net of connections and allows for a more individualized visualization of the dataset with many different levels of zoom. Different Levels of zoom can be seen in Figure 10.

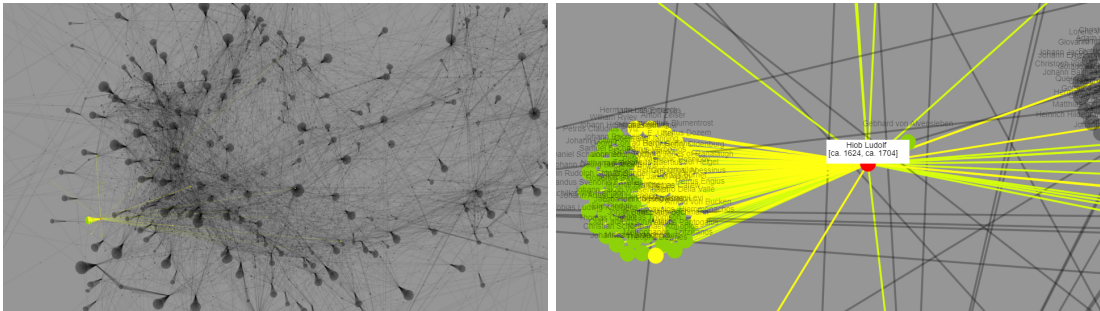


Figure 10: Different zoom levels during graph exploration. In this example Hiob Ludolf has been selected.

Another needed tool is the visualization of all the connections of a single vertex. This may be important as vertices close to each other on the screen may not always be connected even if it appears that way. The set of connected vertices can be computed efficiently as the user selects an originating vertex. Selecting a vertex is done by clicking on it with the left mouse button. Then the selected vertex gets colored in red and the connected vertices get colored in the same color. The vertex is also moved in the middle of the screen. Coloring the connected vertices and edges in the same color improves the visibility. This is needed as the graph can sometimes be dense and not all vertices close to one another are also connected. When a vertex is selected, a search is also performed for the selected person. The search is also intertwined with the graph the other way around as searching for a person also selects the corresponding vertex in the graph. Additionally, the non selected vertices can be made invisible to further improve visibility with the “Toggle view” button as illustrated in Figure 11.

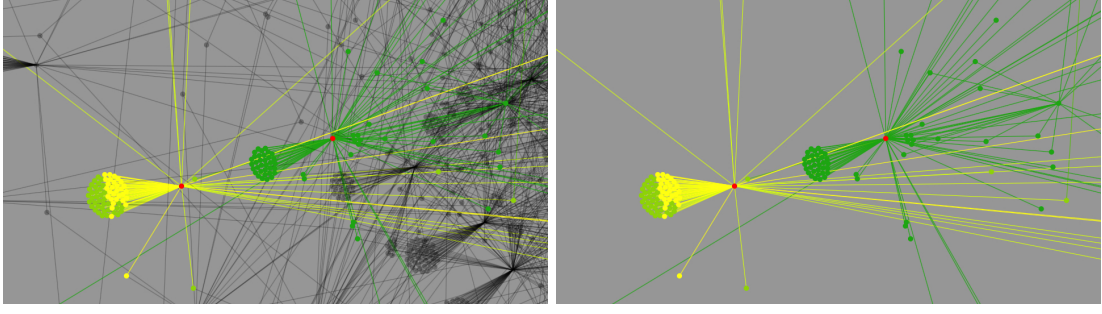
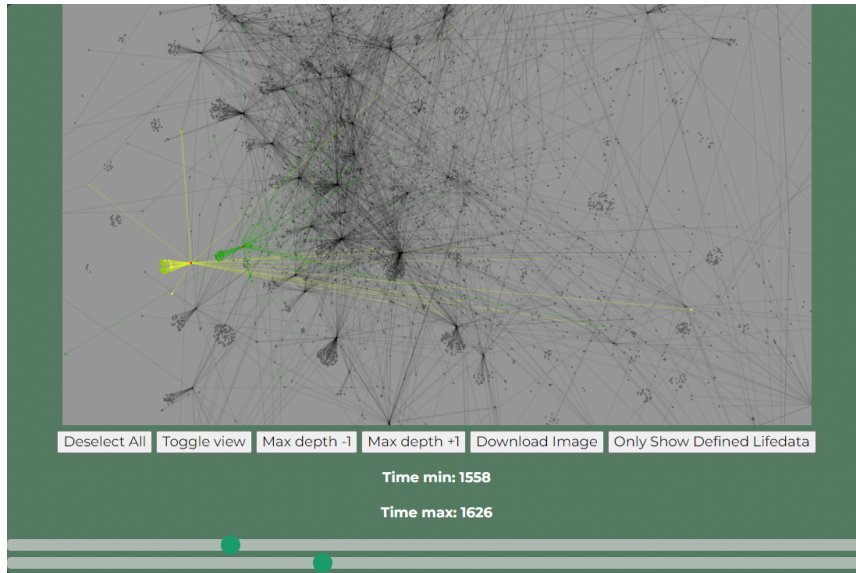


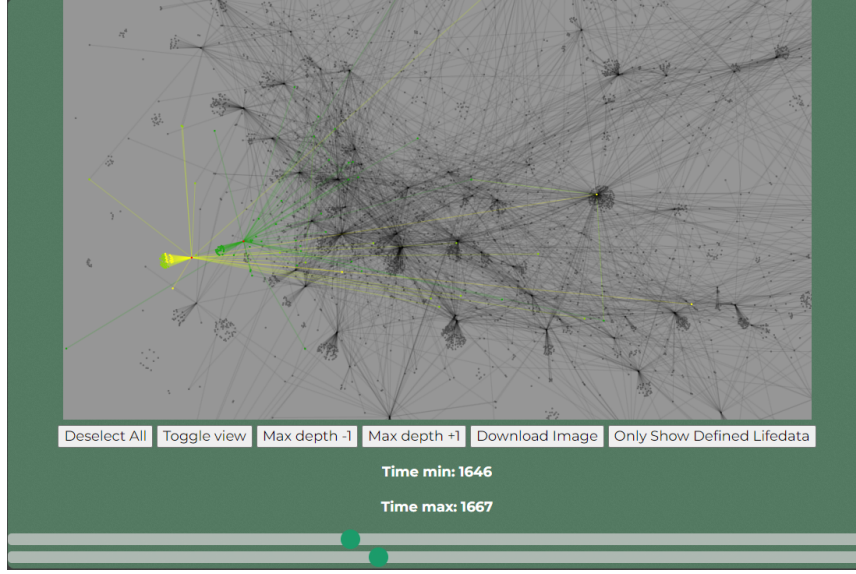
Figure 11: Both images illustrate the graph with two different selected vertices. On the right however the non selected vertices are set to invisible.

When zoomed in, the name of each person appears over the vertex as in Figure 10. The user can then hover over it to also see the birth and death dates if they are available. If only the birth date is available, the death date is computed by adding 70 years to the birth date. The same is done in reverse if only the death date is known. Setting the lifespan in such cases to 70 years may appear optimistic, but it enables a visualization of the evolution of the graph over a large time period. For more detailed information however it may be misleading. One could argue, that some people without either a birth or death date are not important enough and therefore it may be an acceptable assumption to set the dates.

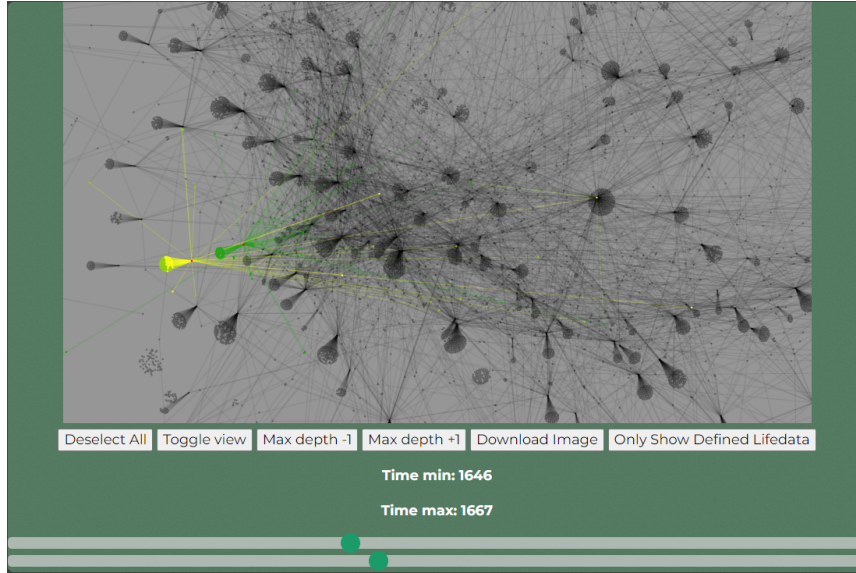
The evolution of the graph over time can also be visualized. By changing the position of two sliders for the minimum or maximum year to display other vertices and edges appear and disappear. If a person lived from 1483 to 1546 for example the person can still be seen if the minimum slider is set to 1546 as the person was still alive in that year. However if it is set it to 1547, the vertex will disappear from our graph. The maximum slider works in the similar where it is filtering for birth dates. In order to only display the people with available birth and death dates it the “Only Show Defined Life Data” button can be toggled. If enabled all vertices with undefined birth and death dates are filtered out. Changing the visible time values is illustrated in Figure 12.



(a)



(b)



(c)

Figure 12: Changing the values on the time sliders changes the currently visible vertices of the vertices. The person selected in green disappears in the last image which indicates that he died before 1732.

Six degrees of separation[5] is the idea that every living person is connected to every other. This chain consists out of at most six friend of a friend connections. As the given network consists out of only links and vertices, and as it is also much older than the social networks in the digital age, more than six degrees of separation may be needed. Another important factor for the given graph is the time scale. While modern social networks existed over a time period over at most 20-30 years, the friendship album network contains entries over a 500 year long interval. In order to explore such overarching connections the depth of a selection can be changed up and down. As the computation gets exponentially more expensive with each level as illustrated in Figure 13, it is not advised to compute more than at most two degrees of separation.

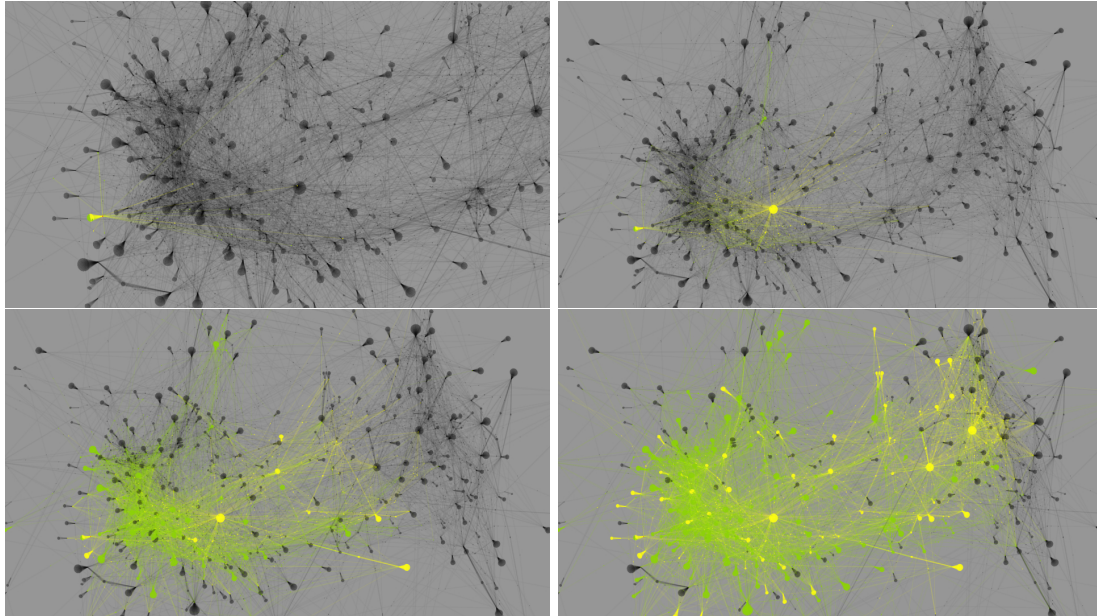


Figure 13: In the first image a depth of one is used and only the direct “friends” of Hiob Ludolf are visible. With each image the depth increases by one for a deepest Level of four.

When multiple people are selected, the colors also mix when they have a “friend” in common.

5.3 Time Diagram

Statistics are very powerful for introducing important aspects of a topic to users. An important aspect for the understanding of how friendship albums were used, is when and how often they were used. Therefore, a time diagram, which can display different kinds of temporal data to the user, was developed. To get all the time data, at first the raw data set needed to be improved. More about this topic can be found in Section 3.4.

The different diagram types can be chosen by the user by clicking at the corresponding button. Directly after the overall tool was started, inside the time diagram therefore the view visible in Figure 14 is presented.

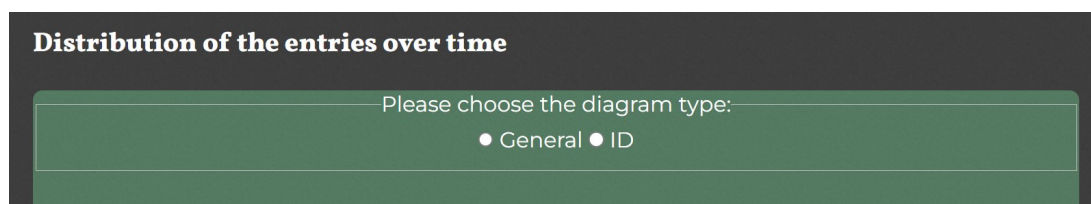


Figure 14: Main View of the Time Diagram

An interesting part of the time data of course is the view over the whole data set with the friendship album entries. Hence, at first the “general” time diagram was developed. As soon as the “general” button is chosen, the user has to decide, if he/she wants to see an overview of the friendship album entries made in one specific year or in a specific year interval (see Figure 15).

Distribution of the entries over time

Please choose the diagram type:

☒ General
☐ ID

Please choose year or interval.

☐ Year
☐ Interval

Figure 15: Main View of the General Time Diagram

By selecting “Year” the default diagram is loaded. The decision of using the year 1724 as default year to display was taken, because in this year the most entries were made overall. The diagram shows the amount of entries made per month inside the selected year (see Figure 16).

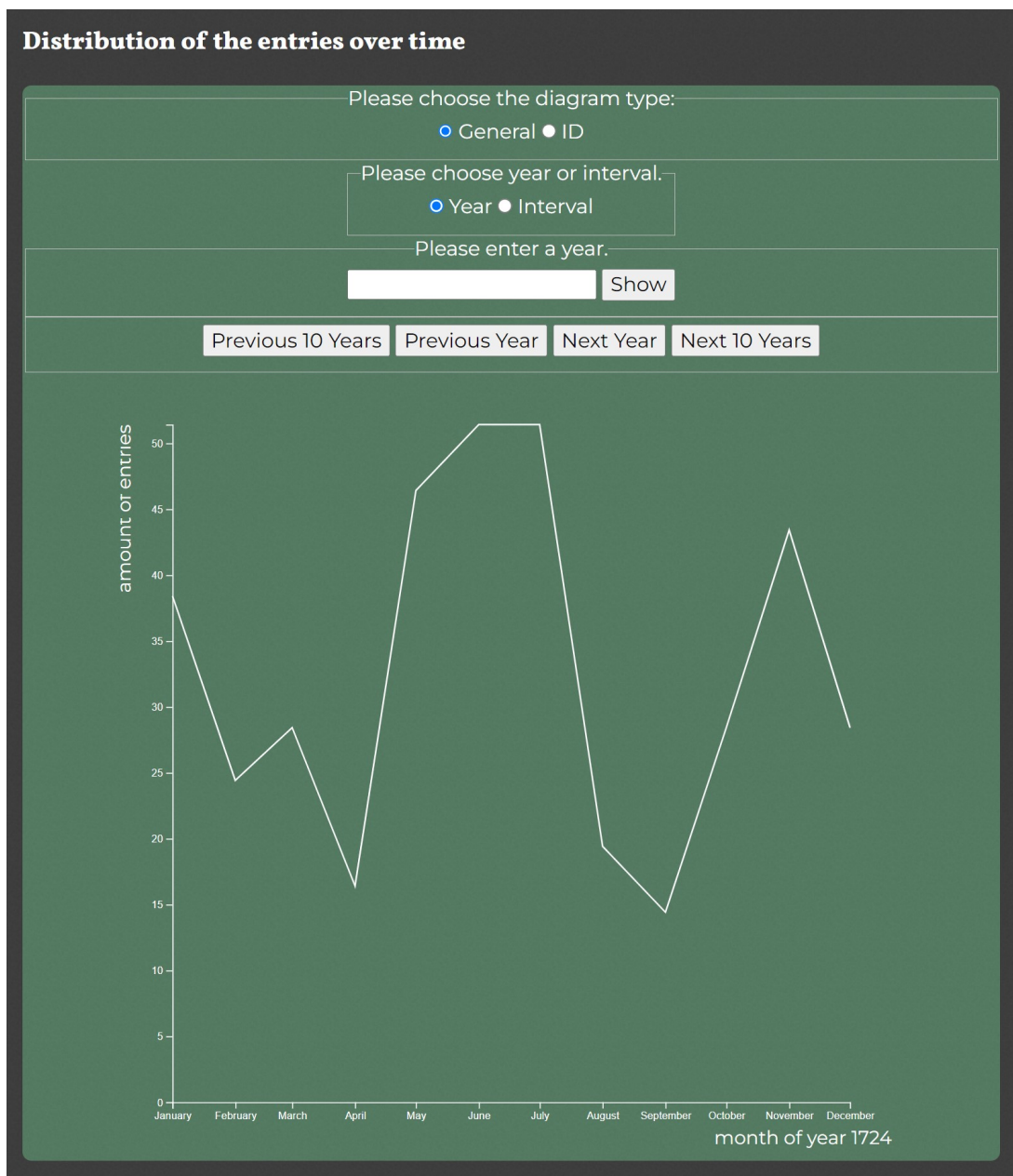


Figure 16: Entry Distribution of the year 1724

If one wants to see the entry distribution for another year, one simply can type the desired year inside the input field and click onto “Show”. The requested diagram will be shown afterwards. Because the majority of the given entries can be located in the 17th and 18th century, it also can happen, that for the selected year no data is available. If this is the case, it is communicated to the user (see Figure 17).

The screenshot shows a web interface titled "Distribution of the entries over time". It has a dark green background with white text and buttons. At the top, it says "Please choose the diagram type:" with two radio buttons: "General" (selected) and "ID". Below that, it says "Please choose year or interval:" with two radio buttons: "Year" (selected) and "Interval". Underneath, it says "Please enter a year:" followed by an input field containing "1800" and a "Show" button. Below the input field are four buttons: "Previous 10 Years", "Previous Year", "Next Year", and "Next 10 Years". At the bottom, a large green box contains the text "There is no data available for the year 1800."

Figure 17: (Not available) Entry Distribution of the year 1800

If a user wants to explore for example, how the amount of entries changed over the years, one can use the buttons provided below the input field. With these buttons it is possible, to jump to the previous or next year or even to the previous or next ten years. The buttons can be used as often as desired.

As already mentioned at the beginning of this section, it is also possible to investigate a year interval. Thus, simply the “Interval” button needs to be clicked. As well as for the “year” diagram a default year interval is loaded afterwards. Due to similar reasons as for the default year view in the before presented diagram, the default interval 1720-1725 was chosen. The corresponding diagram is visible in Figure 18.

Within the scope of the interval display it is also possible to switch between two different views for the x-axis. Per default the monthly view as for the single year representation is shown. Different from the single year representation, here the overall amount of entries is shown for each month. Taking the default interval as an example, the amount of entries within the month July, i.e. contains the amount of entries of the July of each year inside the interval 1720-1725 and therefore the sum of the entries of all these years. To get also an overview, how many entries were made in every year of the interval, the user therefore also can select the yearly view for the corresponding interval. Of course, the interval can be customized by the user as well (see Figure 19).

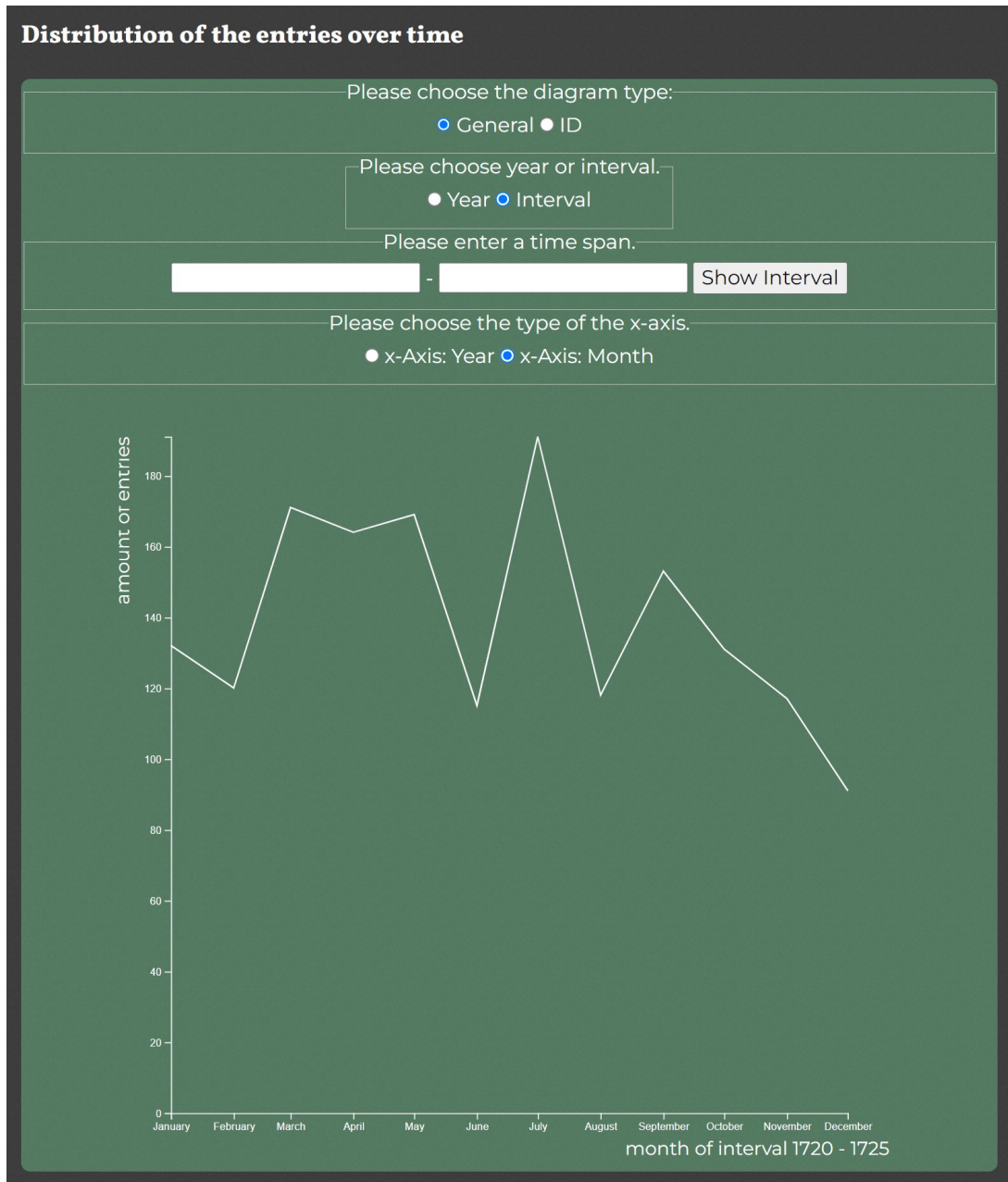


Figure 18: Entry Distribution of the interval 1720-1725 (Monthly View)

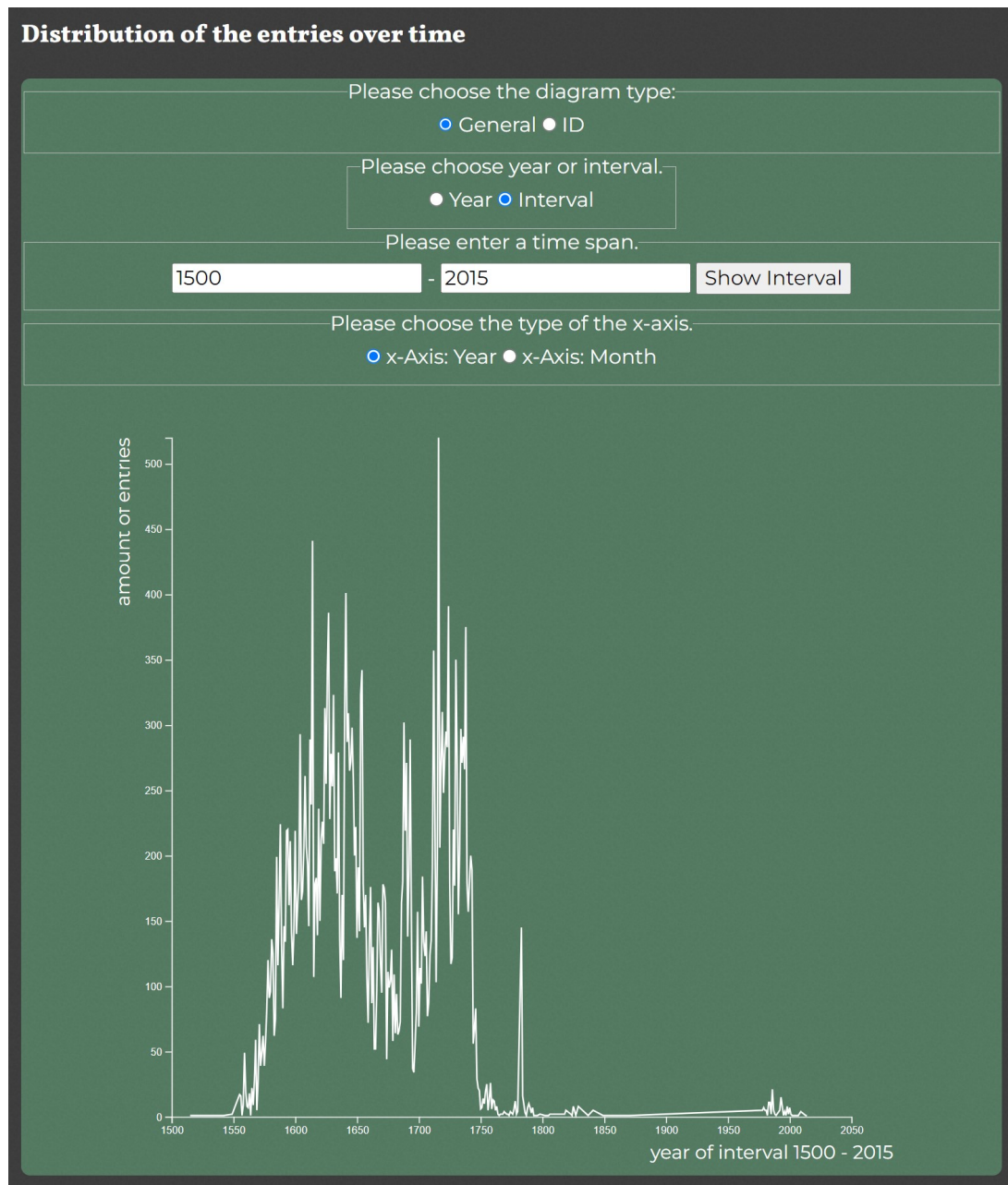


Figure 19: Entry Distribution of the interval 1500-2015 (Yearly View)

Because one of the main goals of this project was to connect different diagrams with each other, another diagram, which connects the time data with single persons, was developed. These single persons can be entry authors, persons who entered in a friendship album, or book owners, persons who owned a friendship album. One single person also can connect both of this possibilities.

To get for example an overview over the entries one specific person made, at first a name or GND-ID needs to be entered into the search bar. If no name or GND-ID was entered into the search bar, a hint explaining the problem is presented to the user. More information about the search bar can be found in Section 5.1. Afterwards, the corresponding entry distribution for the entered person is shown in the “entry author” diagram.

In the following the functionalities of the different diagrams are explained on the person of Hiob Ludolf.

Per default the diagram shows an interval starting with the year the first entry was made by the selected person (1644) and ending with the year the last entry was made (1671). Hiob Ludolf himself did not

enter in many friendship albums (three in total). In Figure 20 it is possible to see the mentioned three entries inside the interval 1644-1671. As explained in Section 3.4 some time information provided only could contain year information. For Hiob Ludolf for all his entries this was the case. This explains the straight line shown in this figure, because no indication about the months could be viewed and therefore the value $\frac{1}{12}$ was added three times for each month.

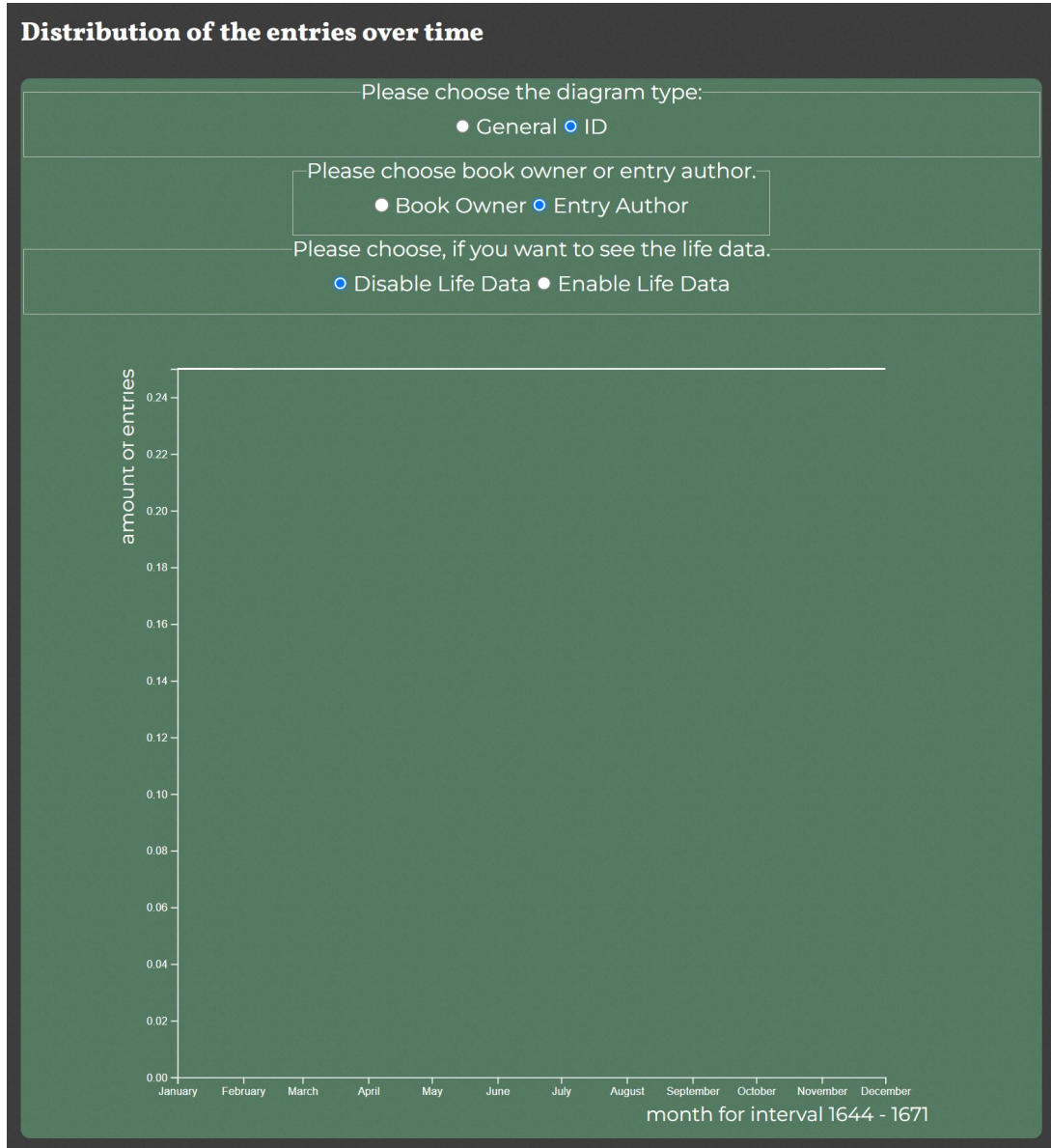


Figure 20: Entry Distribution of entries made by Hiob Ludolf (view without life data)

Inside the “entry author” diagram it is furthermore possible to display the yearly distribution of the entries made. Per default the monthly view is shown like indicated before. Thus, the “Disable Life Data” button is selected. If one wants to show, at which time in the life of the author he/she made his/her entries, one can select the button “Enable Life Data”. As soon as the “enable life data” button is clicked, the x-axis therefore changes to the interval birth year - death year. In the case of Hiob Ludolf, the x-axis shows the interval 1624-1704 (see Figure 21).

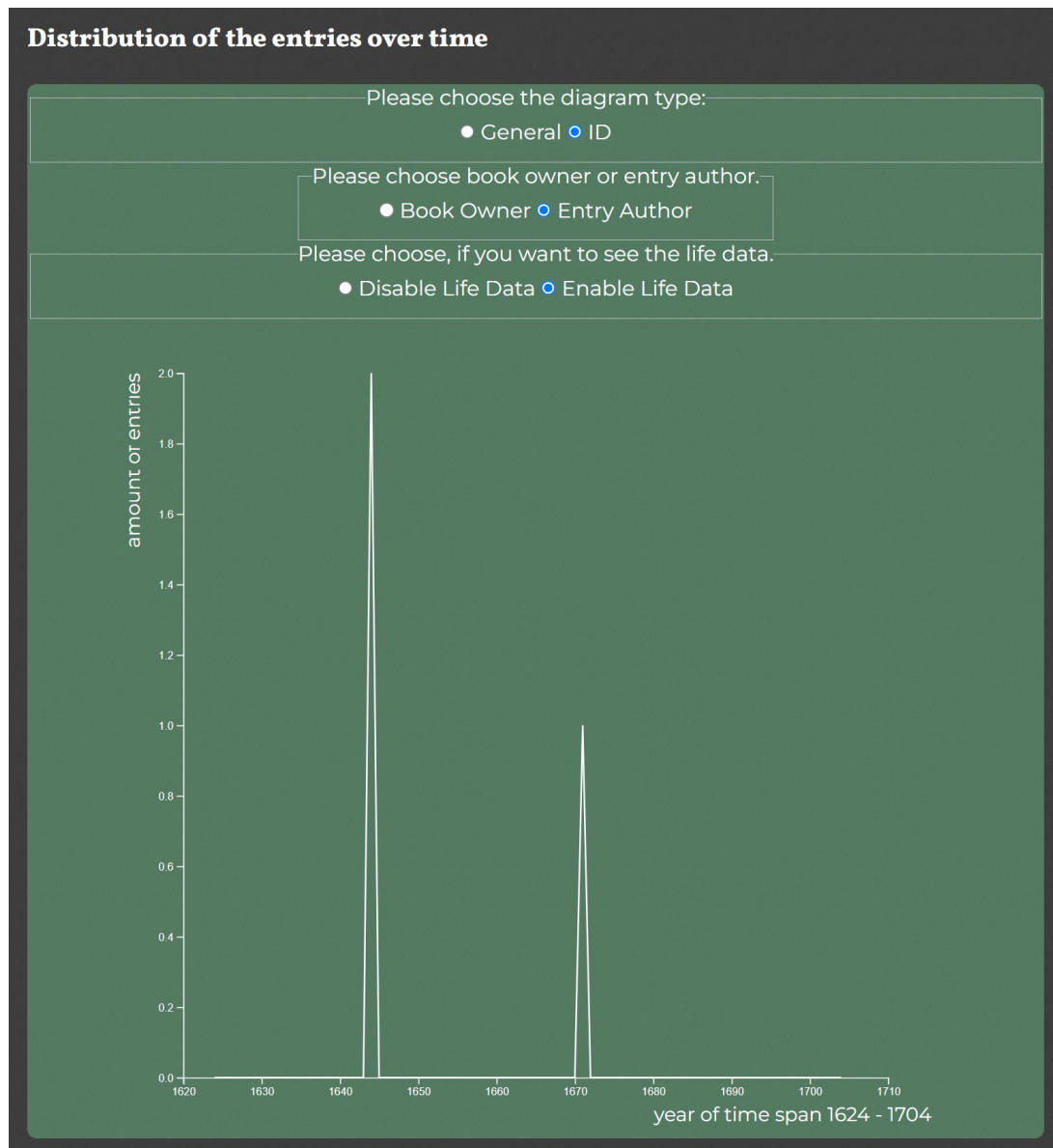


Figure 21: Entry Distribution of entries made by Hiob Ludolf (view with life data)

If the person also has owned a friendship album, the entries of this book can be displayed by switching to the “book owner” diagram. Hiob Ludolf is a prominent example for a person, which owned a friendship album with many entries. In his lifetime he could get 83 entries in total. Hence, if one selects the before mentioned “book owner” button, the x-axis now shows again an year interval starting from the first entry made and ending with the last entry made with the above mentioned monthly display. In this case of Hiob Ludolf the interval now changed again back to 1644-1671. Despite that, the entries of one person also could be made within one year. The interval then would only be a time span of one year.

Also for the “book owner” view it is of course possible to consider the yearly view inside the life data of the book owner (here again 1624-1704). The mentioned view for Hiob Ludolf is visible in Figure 22.

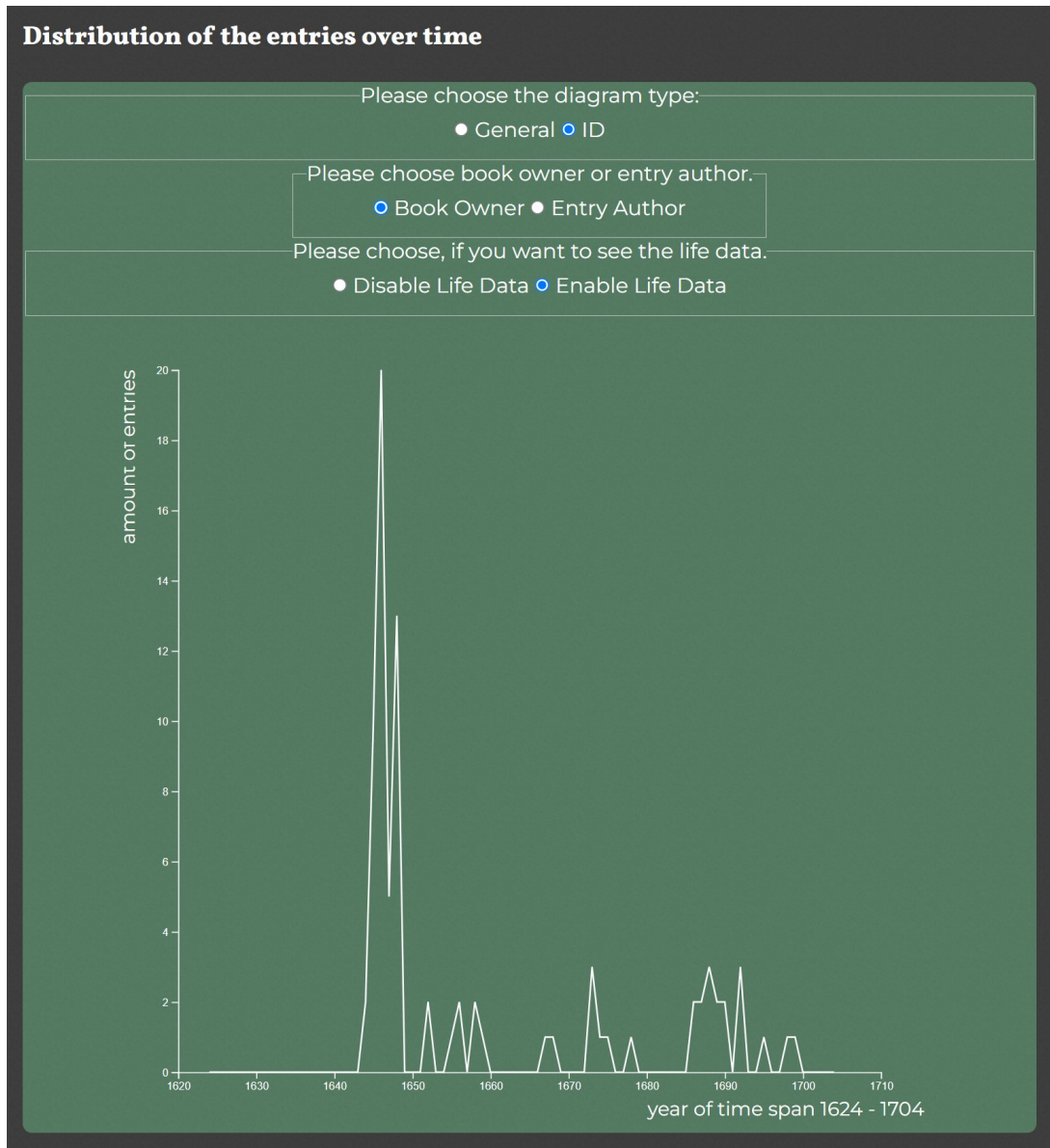


Figure 22: Entry Distribution of the friendship album owned by Hiob Ludolf (view with life data)

5.4 Language Diagram

One of the more interesting data's to look at are the languages, in which the entries are written, as the language could indicate the cultures and ethnicities of the persons mentioned in the given data set.

The approximately 34.000 entries inside the provided data set overall are written in 34 different languages ranging from the more common ones like Latin (27.549), German (8.797), French (2.460) and Greek (3.077) to more exotic ones like Arabic (125), Danish (4) or Amharic (1).

Of course, a graph with a wide span of language frequencies would be very incomprehensible, wherefore the decision of just putting the six most common languages on one graph was made (see Figure 23).

For creating this bar chart, at first a script, which filters out the language column from the friendship album data set and counts how often each language appears, was written. During this implementation the first problem occurred: Not every entry was written in only one language - on the contrary, many entries were written in multiple languages.

As already visible in Figure 23 it was possible to use this problem and transfer the first graph into a

stacked bar chart where each colour and each bar represents one language. Each bar displays the amount of entries the language had at all and each colour in the stack represents the amount of entries the “bar language” shares with the “colour language”. For example, the orange language represents Latin. Therefore, one can see, that Latin shares around 20.000 entries with Latin, which simply means, that 20.000 entries were written in purely Latin. Another aspect of this bar is, that Latin shares also around 3.000 entries with German, which can be figured out by looking at the orange part from the German bar or by looking at the yellow part from the Latin bar.

Of course, it can be quite hard reading the chart, especially if it is not clear, which language is represented with which color. This is the reason, why a tool tip, which shows the language, that is currently hovered over with the mouse, was implemented. Additionally, within the tool tip also the amount of entries written in the specific language is visible as “Value” (see Figure 24).

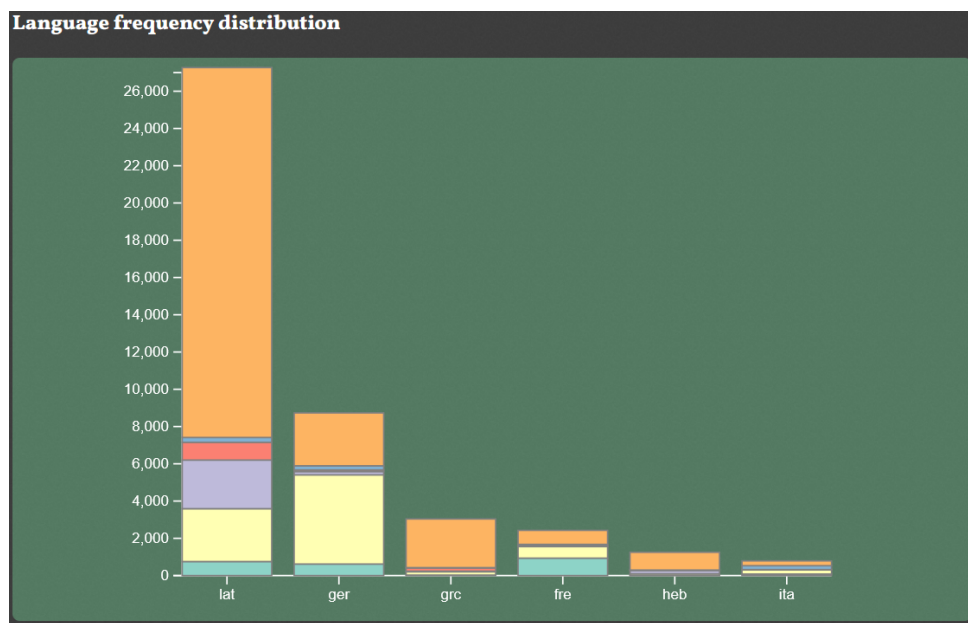


Figure 23: The six most common languages

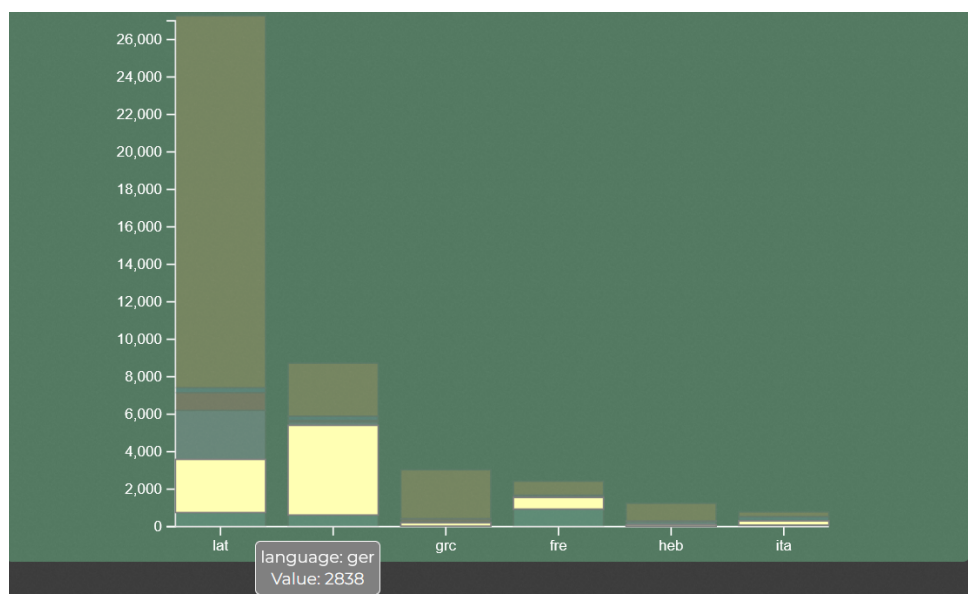


Figure 24: The six most common languages with a tool tip about German

6 Heuristic Evaluation / Expert Review

In order to assess the effectiveness of the developed visualization and their interfaces, the aspired effect for each component of the prototype was reflected in a set of hypothesis. Those hypothesis build a guideline for an user interview where two experts from the “Herzogin Anna Amalia Bibliothek” were asked to try out the prototype and openly reflect about it’s usability. The two experts were Katja Lorenz, head of special collections, and Stefanie Kruse, librarian.

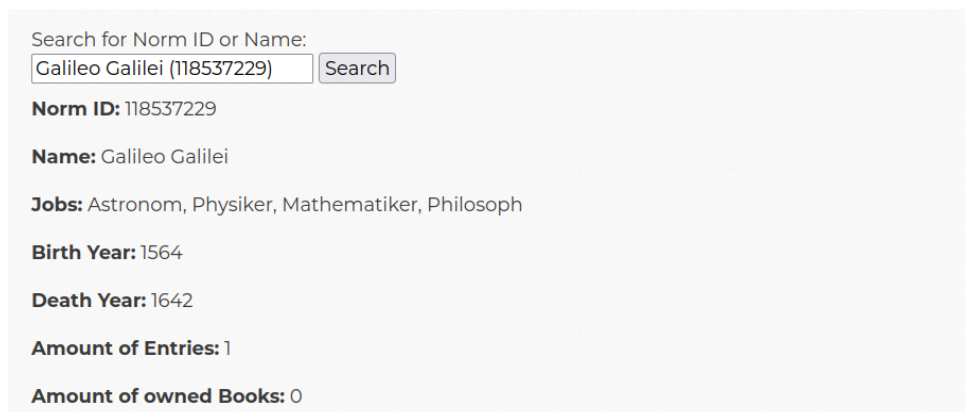
6.1 Search

Hypothesis 1: *By using the search, one can effectively search for specific people by name or GND-ID.*

The experts are able to search for specific people in the dataset, even if other people with a similar name exists. This is possible because of the GND-ID that is displayed next to the person. However, it is noted by the experts that the ID can not intuitively be recognized as a GND-ID, it is suggested to add the standard prefix for GND-IDs, in explanation write “gnd/118537229” instead of “118537229”.

Hypothesis 2: *The search displays relevant information about the person.*

All attributes chosen for display were regarded as helpful. In addition to the currently selected attributes the experts were also interested in the region connected to the person. This information is available in the GND record but not used for display right now.



The screenshot shows a search interface with a text input field containing "Galileo Galilei (118537229)" and a "Search" button. Below the search bar, the results are displayed in a list-like format with the following attributes:

- Norm ID:** 118537229
- Name:** Galileo Galilei
- Jobs:** Astronom, Physiker, Mathematiker, Philosoph
- Birth Year:** 1564
- Death Year:** 1642
- Amount of Entries:** 1
- Amount of owned Books:** 0

Figure 25: Performed search during the user study

6.2 Graph

Hypothesis 3: *Navigating through the graph and selecting people is intuitive and effective.*

The experts struggled to navigate through the graph. One important reason for that is the low contrast ratio between the color of the nodes, edges and the background. It was proposed to give users a choice for the background color. Also interface elements like the “Only Show defined Life Data” or the “Max depth” buttons can’t be understood intuitively and required further instruction during the interview. It was also asked if the “Only Show defined Life Data” button is useful in different contexts apart from the combination with the time sliders. The feature to download the current view as an image is appreciated as a very helpful tool for researchers that want to document their research.

Hypothesis 4: *Certain time periods can be adjusted by the sliders, so that it is possible to quickly filter by time period.*

For the experts it seems counter intuitive to use two sliders to select a time window. The experts

raise the question if two sliders above each other are necessary and for which reason. Before clicking on both sliders it is not possible to tell their function.

Hypothesis 5: *The graph view offers a novel approach to explore the friendship albums.*

The experts reflect that such a possibility to explore their friendship album data is not yet available to researchers. The graph visualisation offers a unique mode of exploration that can potentially answer certain research question more effective and certainly allows to formulate different research question when using the visualisation.

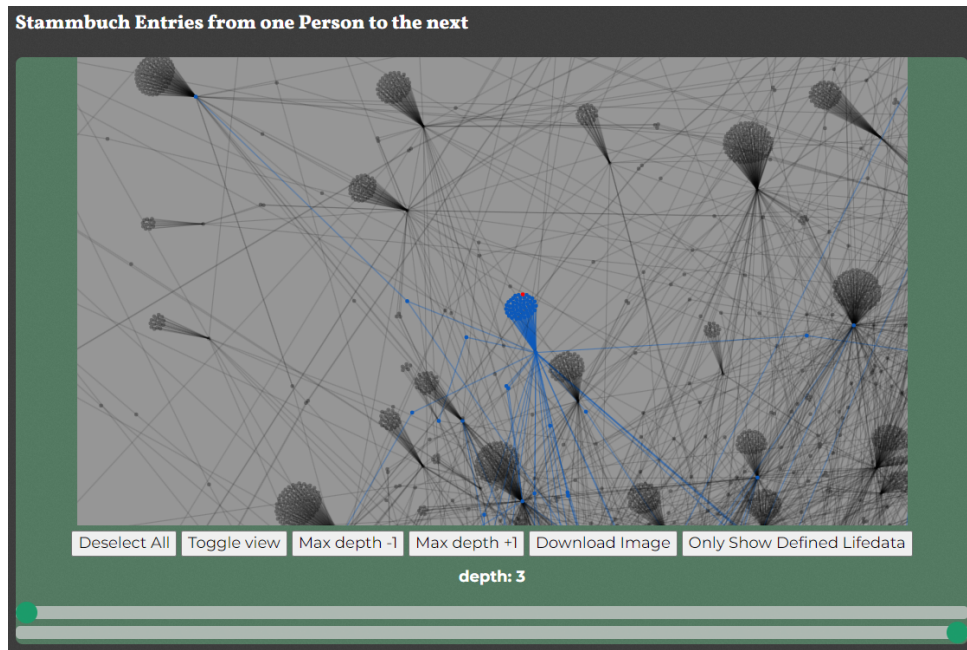


Figure 26: Performed graph query during the user study

6.3 Time Diagram

Hypothesis 6: *The time diagram for individual persons in combination with the search bar offers the possibility to examine individual entries and family books in more detail.*

The experts are able to use the time diagram to derive more complex analysis about the searched person. It is noted that fractional numbers were not expected on the y-axis and the experts had no intuitive understanding why e.g. 0.8 friendship album entries in March are possible. Also it is noted that the date of a written entry is a point event, therefore interpolating a linear curve over the points seems off to the experts.

Hypothesis 7: *The function of displaying the life data of the corresponding person within the person-related time diagrams enables a more detailed investigation of the connection of the person with the entries.*

The experts find this function useful and speculate about the possibility to combine this visualisation with data of other publications of this person or special events in their life.

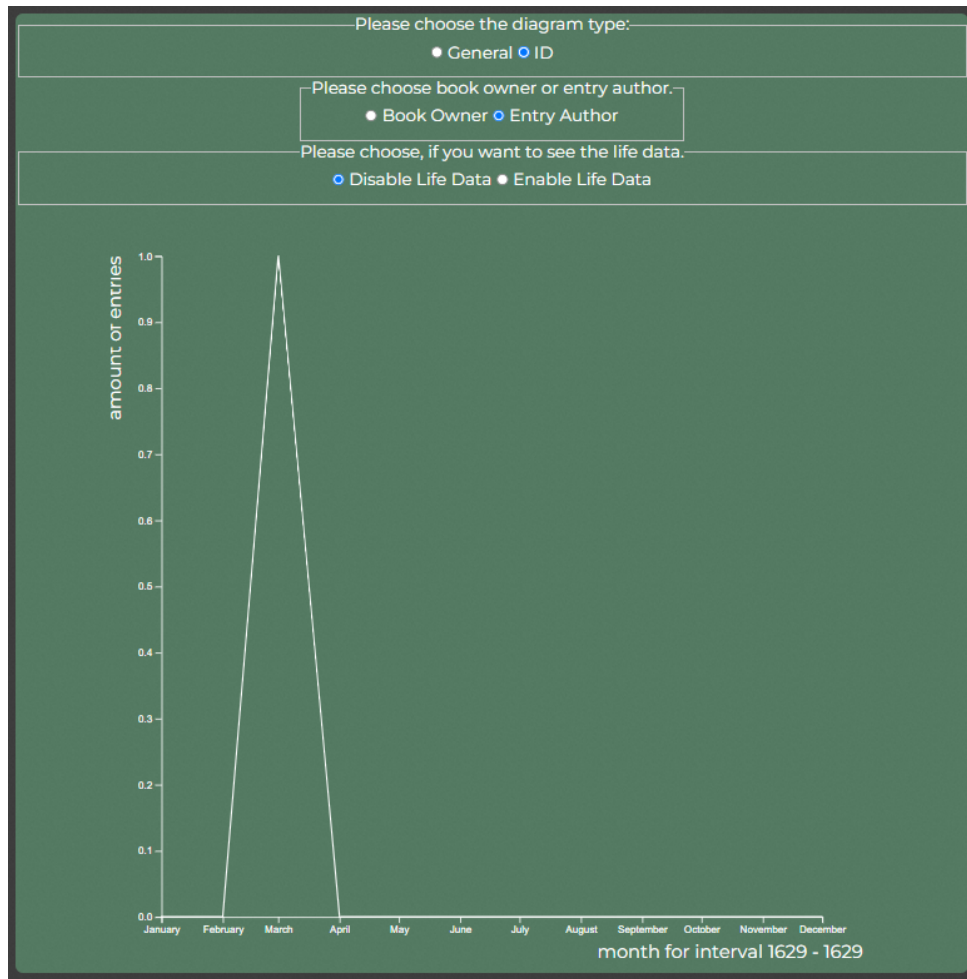


Figure 27: Performed query on the time diagram during the user study

6.4 User Study Conclusion

All in all the experts were very satisfied with the new possibilities to discover the friendship album data set offered by the prototype. Helpful feedback on how to improve the user interface was further collected. In general it has to be stated that in its current form the prototype was not responsive. Additional potential for optimizing the JavaScript code must be explored before the prototype can be used by the public.

7 Challenges & Solutions

In summary, this project has highlighted several critical challenges inherent in the realm of friendship albums, exposing the intricate nature of historical documents in general. Those were, of course, not intended to be machine-readable and therefore inconsistent with missing data points like the date below each entry. The non-standardized varieties of friendship albums, marked by diverse and evolving formats across centuries, poses a significant obstacle to uniform digitization efforts. The attempts at digitization, exemplified by initiatives such as GND DNB/DND, have encountered hurdles, particularly in dealing with incomplete and non-standardized data. Notably, the temporal dimension of entries introduces complexities, as seen in inconsistent time-related data. Jun-Prof. Franziska Klemstein (“Digitale Methodik in den Geistes- und Kulturwissenschaften”, Hochschule Mainz) explained to us that especially joint projects between libraries and archives like the GND are not always able to ensure a consistent degree of quality which makes them an unreliable source for further research.

Based on the GND records, the merging of occupation information within biographical data to social groups adds another layer of complexity to the data preparation process. The limited knowledge about

archaic professions raising questions about the accuracy of attributions, such as distinguishing between a doctor, teacher, or nurse. The project also touched on the difficulty in detecting and digitizing historical text. Which itself adds numerous challenges.

In light of these challenges, this project recognizes the significance of addressing user expectations and needs in navigating and utilizing the digitized friendship album records. It is important to note that also in the context of friendship albums different groups of users have different requirements. The user study conducted within this research becomes essential in understanding user preferences and expectations, providing valuable insights into what users desire to see and how they intend to engage with the digitized friendship album entries.

8 Conclusion

During the implementation of the interactive tool many issues have arisen like already mentioned in Section 7. It was possible to create different visualizations for the data given at the beginning of this project. Also the goal of connecting this different visualizations with each other interactively could be partly achieved. A graph, which shows connections between the different persons mentioned in the data set was developed as well as an interactive search bar and a time series diagram. Unfortunately it was not feasible to develop a visualization regarding the social groups the entry authors and album owners belonged to. Nevertheless, the tool prototype could be tested within a small expert review.

9 Future Work

Building upon the insights gleaned from the challenges and discoveries of this project, several avenues for future exploration and development emerge. One key direction involves the generation of additional visualizations, specifically delving into the diverse complexity of professions and occupations documented in the GND records. Beyond the conventional visual representations employed in this study, there is potential for exploring alternative diagram types, such as Chord Diagrams, which could provide a more nuanced and interconnected portrayal of relationships between professions or social groups within the historical context.

Moreover, the development of a social groups diagram stands out as a promising area for future exploration. Although not extensively pursued in this project due to challenges related to accurate attribution, dedicating focused efforts to address these attribution issues could unlock valuable insights into the social dynamics embedded in friendship albums. This would then require to make large use of the proposed statistical hypothesis testing framework in order to achieve a high degree of assurance during development.

Looking ahead, there is also the prospect of envisioning an interactive website tailored for the German National Library (DNB) or the Herzogin Anna Amalia Bibliothek (HAAB). Such a platform could serve as a dynamic repository for digitized friendship album entries, allowing users to engage with the historical content in an interactive and user-friendly manner. Incorporating features like advanced search functionalities, interactive visualizations, and customizable displays could enhance the user experience, addressing the complexities outlined in this study. This project can function as an example on how the proposed interaction paradigms can be beneficial for such a platform in the future.

Additionally, the exploration of advanced machine learning or natural language processing techniques holds promise for improving the accuracy and efficiency of text detection and digitization. As well as advanced methods of computer vision can be incorporated to make the content of the entries themselves machine accessible. Future work in this domain could focus on refining existing methodologies or exploring novel approaches to overcome the here mentioned challenges associated with handwritten historical texts.

10 Bibliography

- [1] *Design von reCAPTCHA anpassen*. 2023. URL: <https://developers.google.com/recaptcha/old/docs/customization?hl=de>.
- [2] *How the Google Books team moved 90,000 books across a continent*. 2023. URL: <https://blog.google/products/search/google-books-library-project/#:~:text=Explore%20Google%20Books%20today%2C%20and,be%20searched%20and%20accessed%20online..>
- [3] DATAtab Team. *Test auf Normalverteilung*. Last accessed: 25 August 2023. 2023. URL: <https://datatab.de/tutorial/test-auf-normalverteilung>.
- [4] DATAtab Team. *Parametrische und Nichtparametrische Tests*. Last accessed: 25 August 2023. 2023. URL: <https://datatab.de/tutorial/parametrische-und-nichtparametrische-tests>.
- [5] *Six degrees of separation*. 2023. URL: https://en.wikipedia.org/wiki/Six_degrees_of_separation.

11 Appendix

Our categorisation (jobs to social groups):

Weltliches Akademisches Umfeld

1. Studenten und Schüler
 - (a) Student (1714)
 - (b) Jurastudent (835)
 - (c) Medizinstudent (114)
 - (d) Schüler (53)
2. Leitende Akademische Berufe
 - (a) Rektor (704)
 - (b) Konrektor (68)
 - (c) Schulleiter (63)
3. Lehrende Akademische Berufe
 - (a) Hochschullehrer (3513)
 - (b) Gymnasiallehrer (1195)
 - (c) Lehrer (263)
 - (d) Pädagoge (256)
 - (e) Gelehrter (76)
4. Humanwissenschaftliches Umfeld
 - (a) Allgemein
 - i. Humanist (52)
 - (b) Jura
 - i. Jurist (4230)
 - ii. Rechtsanwalt (115)
 - iii. Syndikusanwalt (50)
 - (c) Geschichte und Philosophie
 - i. Historiker (567)
 - ii. Philosoph (847)
 - (d) Sprache und Kultur
 - i. Philologe (600)
 - ii. Klassischer Philologe (309)
 - iii. Übersetzer (62)
 - iv. Gräzist (59)
 - v. Hebraist (149)
 - vi. Orientalist (306)
 - (e) Andere
 - i. Polyhistor (149)
5. Naturwissenschaftliches Umfeld
 - (a) Allgemein
 - i. Naturwissenschaftler (112)
 - (b) Medizin
 - i. Arzt (1714)

- ii. Leibarzt (121)
 - iii. Apotheker (103)
 - iv. Anatom (67)
- (c) Mathematik, Physik, Biologie, Chemie
 - i. Mathematiker (407)
 - ii. Physiker (194)
 - iii. Astronom (116)
 - iv. Botaniker (140)
 - v. Chemiker (89)
- (d) Andere
 - i. Geograf (51)

6. Andere

- (a) Magister (107)
- (b) Respondent (89)
- (c) Genealoge (54)

Kirchliches Umfeld

1. Theologie

- (a) Theologe (2213)
- (b) Theologestudent (1019)
- (c) Pfarrer (3747)
- (d) Evangelischer Theologe (2213)
- (e) Professor für Theologie in Altdorf (54)

2. Kirchliches Umfeld (ausgeschlossen Theologie)

- (a) Prediger (398)
- (b) Hofprediger (336)
- (c) Propst (143)
- (d) Pietist (83)
- (e) Kantor (82)
- (f) Kirchenlieddichter (303)
- (g) Geistlicher (430)
- (h) Konsistorialrat (136)
- (i) Abt (59)
- (j) Superintendent (783)
- (k) Generalsuperintendent (293)

Bürgerliches Umfeld

1. Beamtenberufe

- (a) Beamter (218)
- (b) Amtmann (81)
- (c) Kämmerer (73)
- (d) Hofbeamter (67)

2. Berufe im Buchhandel

- (a) Bibilothekar (491)

- (b) Verleger (119)
- (c) Selbstverleger (111)
- (d) Buchbinder (108)
- (e) Drucker (71)
- (f) Publizist (57)

3. Andere

- (a) Kupferstecher (79)

Politisches und Militärisches Umfeld

1. Offiziere und Soldaten

- (a) Militär (157)
- (b) Offizier (109)
- (c) Soldat (53)
- (d) General (50)

2. Politiker

- (a) Politiker (398)
- (b) Bürgermeister (238)
- (c) Kanzler (143)
- (d) Rat (87)

3. Höfisches Umfeld

- (a) Hofrat (201)
- (b) Hofmeister (87)

4. Diplomaten

- (a) Diplomate (217)
- (b) Gesandter (68)

5. Andere

- (a) Patriziat (125)
- (b) Geheimer Rat (193)

Künstlerisches und Musikalisches Umfeld

1. Kunst

- (a) Maler (125)
- (b) Zeichner (74)

2. Schrift

- (a) Schriftsteller (688)
- (b) Dichter (69)

3. Musik

- (a) Komponist (108)
- (b) Musiker (87)