universität wien

MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

The BAJer

A tool for the analysis of astronomical journal bibliometrics

verfasst von / submitted by

Aaron Lauterer, B Sc

angestrebter akademischer Grad / in partial fulfillment of the requirements for the degree of

Master of Science (M Sc)

Wien, 2020 / Vienna, 2020

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	A 066 861
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Masterstudium Astronomie
Betreut von / Supervisor:	Univ.-Prof. João Alves

The BAJer

A tool for the analysis of astronomical journal bibliometrics

Aaron Lauterer, BSc

Abstract

Aaron Lauterer, BSc

The Journal Impact Factor (JIF) is being questioned as the controlling metric to determine a journal's ranking within the scientific community. Astronomy is in the unique situation of having the Astrophysics Data System (ADS), which is a freely accessible data base that tracks publications in Astronomy with its associated metadata. ADS' existence allows for an exploration into the papers' bibliometrics from the Astronomy community openly and transparently. While the ADS offers an in-depth search interface, it is difficult to use on large data-sets when looking at publication output of multiple journals over a considerable time period. Additionally, the data has noise, and some critical metadata is *hidden* within the data. This Master thesis aims to create a software package that makes the bibliometric exploration of ADS simpler. The software, that I name *The BAJer*, regularly downloads publication data from the ADS for a configurable list of journals. *The BAJer* cleans the data from noise and extracts *hidden* data. The cleaned and extended data is then offered in an already aggregated form via a REST API to be consumed by clients. A web-based client is the last part of the package to have easy and direct access to the aggregated data and visualizations thereof. The larger goal of this thesis is to establish the groundwork and tools that will allow for the exploration and better understanding of different publication metrics, informed by a multidimensional view of the data.

Key words: bibliometrics, data analysis, visualization, software

Zusammenfassung

Der Journal Impact Factor (JIF) als dominante Metrik um Journale zu bewerten wird in der wissenschaftlichen Gemeinschaft in Frage gestellt. Die Astronomie ist in der glücklichen Lage das Astrophysics Data System (ADS) zu haben. Es ist frei zugänglich und verfolgt Publikationen in der Astronomie und die damit verbundenen Metadaten. Die Existenz des ADS erlaubt es die Bibliometrie der von der astronomischen Gemeinschaft veröffentlichten Publikationen in einer offenen und transparenten Weise zu erforschen. Auch wenn das ADS eine tiefgehende Suche ermöglicht, ist es doch schwer zu verwenden, wenn man große Datensätze wie die gesamte Publikationshistorie von mehreren Journalen über einen langen Zeitraum untersuchen möchte. Zusätzlich haben die Daten Rauschen und einige kritische Metadaten sind in den Daten *versteckt*. Diese Masterarbeit hat zum Ziel ein Softwarepaket zu erstellen, dass die bibliometrische Erforschung des ADS vereinfacht. Die Software, die ich *The BAJer* nenne, ladet regelmäßig die Publikationsdaten für eine konfigurierbare Liste an Journalen aus dem ADS. *The BAJer* säubert die Daten von Rauschen und extrahiert *versteckte* Daten. Die gereinigten und erweiterten Daten werden in einer schon aggregierten Form über eine REST API den Clients zur Verfügung gestellt. Ein web basierter Client ist der letzte Teil des Softwarepakets um einen einfachen und direkten Zugang zu den aggregierten Daten und deren Visualisierungen zu erhalten. Das langfristige Ziel dieser Arbeit ist es, die Grundlagen und Werkzeuge zur Verfügung zu stellen um die weitere Untersuchung und ein besseres Verständnis von Publikationsmetriken, durch einen multidimensionalen Blick auf die Daten, zu ermöglichen.

Declaration

I hereby declare that this thesis is my own work and effort and that it has not been submitted anywhere for any award. Where other sources of information have been used, they have been acknowledged.

Vienna, September 2020 Aaron Lauterer

Contents

CHAPTER 1

Introduction

1.1 Motivation

There are many ways to measure and rank a journal's performance. At the moment the Journal Impact Factor (JIF) (Garfield, 1955) is the universally used metric upon which a journal is ranked.

The JIF is calculated in the following way; the citations received in a given year y for papers published in the two years prior are put into relation to the number of papers published in the two years prior.

$$\mathrm{JIF}_y = \frac{\sum_{y-2}^{y-1} \text{Citations}}{\sum_{y-2}^{y-1} \text{Published papers}} \tag{1.1}$$

It has come under more scrutiny in recent years (Garfield, 2006) ('Time to remodel the journal impact factor' 2016) (Antonoyiannakis, 2020) (Chawla, 2020) and alternatives have been proposed (Larivière et al., 2016). This should be reason enough to take a look at other metrics of journal publications to get a better understanding. One person who has done this for decades is Helmut A. Abt. He published a wide variety of papers and articles dealing with topics such as increased output of astronomical literature ((Abt, 1998), (Abt, 2010)) as well as questions regarding the citing and referencing within astronomical journals (Abt, 2018), (Abt, 2009).

The main challenge when doing bibliometric research is to find a reliable data source. Commercial providers of JIF data which track publications and citations do not give away such data easily. The field of Astronomy is in a unique position among the natural sciences, as it has a single *authoritative* database that tracks publications and associated metadata, the *Astrophysics Data System*[1] (ADS). It has another important property, everyone can access the data freely as it is not hidden behind a paywall.

ADS offers an in-depth search syntax which works well if one is interested in data for specific authors or institutions. While it can deliver big data-sets easily it is not well suited to do more intricate analysis through its web interface. Then there is noise in the data itself, for example, publications that are tracked but are of no interest when trying to determine the scientific performance of a journal, like editorials or errata.

1 http://adsabs.harvard.edu

Other data is not easily accessible and *hidden* within other data. For example, the country in which an author wrote a paper is not stored in a field on its own. It can, however, be present though as part of the affiliation field, which is a free form text field. Being a free form text field results in various problems such as typos or inconsistent ways of formatting the information, making country information extraction problematic.

To get a better understanding of the bibliometric data in ADS, and how to explore it to compare the various possible publication metrics, a tool is needed that will make this data easily available, minimizes the noise in the data, and reveals important hidden data.

1.2 Goal

The goal of this thesis is to create a software package that will download bibliometric data for astronomical journals from the ADS at regular intervals. The tool should clean the data from noise and create a set of reproducible data visualizations. The larger goal of this thesis is to establish the groundwork and tools that will allow for the exploration and better understanding of different publication metrics, informed by a multidimensional view of the data.

The focus of the thesis is in particular on data cleaning, not available anywhere else at the moment as well as data visualization. The resulting software package should be split into logical units so future users can choose which parts to use into their workflow. For easy accessibility the user interface should be a web-based dashboard that functions with any modern web browser.

1.3 Name

The name “The BAJer” is a wordplay with the abbreviation BAJ, Bibliometrics for Astronomy Journals, and the animal that is used in the logo, a badger.

CHAPTER 2

Methods

During the development of the thesis the following astronomical main journals were used: "Astronomy and Astrophysics" (A&A), "The Astronomical Journal" (AJ), "The Astrophysical Journal" (ApJ), and the "Monthly Notices of the Royal Astronomical Society" (MNRAS).

2.1 First steps

At the beginning of this project I downloaded a first data-set from the ADS to get an idea of what the data looks like. I wrote two small programs, one to fetch the data from the ADS and store it in CSV[1] format called *ads-fetch*[2]. The other is to store the data from the CSV file into an SQLite[3] database called *csv_to_sqlite*[4]. The insight from this early step was important as it helped to create a prototype with basic functionality.

I discovered quickly that the data is quite inconsistent in its quality. The oldest data is from almost two hundred years ago, 1827. The way journals do their publications obviously changed during this time and historical old data is by far not as complete as recent one. When dealing with the disparities, a definition needs to be applied as to when data is considered *bad data* which should not be used.

2.1.1 Definition of *bad data*

Any publication that was erroneously entered in the ADS, either duplicated or incomplete, and is of no bibliometric interest shall be considered *bad data*.

Of no bibliometric interest are for example publications like:

- errata or addenda
- editorials
- other kind of front-end matters, like notices or even obituaries

2.1.2 Other Thoughts on Data

Data that is fetched from the ADS but currently not used in any analysis are keywords and author names. Both suffer from ambiguities. For authors names the way initials and diacritics are handled can lead to many ways of writing the same name. This situation

1 Comma Separated Values
2 https://gitlab.com/aaronlauterer/ads-fetch
3 https://sqlite.org
4 https://gitlab.com/aaronlauterer/csv_to_sqlite

‘hopefully will get remedied in the future by the introduction of an author ID system (like, for example, ORCID).’(Henneken et al., 2015, p. 4)

Keywords have the problem that they are free text. This results in many ways to organize them. Typographic errors are occasionally present too, which complicates the situation.

The quality of the data needs to be considered if keywords or author names will be used for some analysis in the future. Rigorous normalization could be one way to deal with it. The other option would be fuzzy string matching / searching.

During the work another part of problematic data became visible. Looking back in time, in the early 1970s keywords start to disappear completely and in the late 1960s the number of papers with the first author available diminishes greatly. This situation is discussed in more detail in section 2.8 on page 16.

2.2 Architecture

I deemed a server–client architecture to be the best approach to split up the different parts of the software package into their logical tasks. The server side consists of three parts. The core around which everything is built is a relational database. It is fed by the importer which fetches the relevant data from the ADS. The data is validated and cleaned during the import phase. A more detailed explanation of that process is discussed in section 2.5.3 on page 8 and section 2.5.4 on page 9

On the other side of the database is the API–Server. It has only read access and is a very thin layer that translates incoming requests to SQL statements and returns the results to the clients.

Finally, there is the client. This is the software the end user interacts with. A web based client has been created in the context of this thesis, though the API can be used by other type of clients as well. Such could be a processing pipeline which is part of another bigger project.

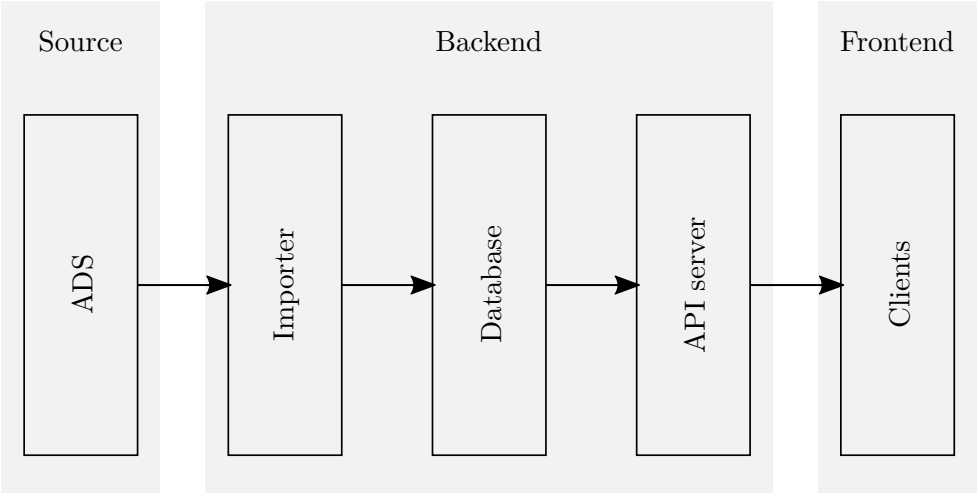


Figure 2.1: Architectural overview of the building blocks of The BAJer

2.3 Implementation

The source code of all software packages discussed in this chapter is publicly available.[1]

Therefore, I will not go too much into code details but give an overview of how the software works and how the different parts interact with each other. If the reader is interested in the actual code I recommend to check out the repositories and look at the code itself. It should be noted that all checks were created with the best effort in mind. When importing several hundred thousand papers (see Table 2.4), some false positives are to be expected and cannot all be handled manually.

2.4 Database

The database forms the core of the system around which the other units align. Thus, it is discussed first.

A relational database has been chosen for the following reasons. The data is very structured in its nature. Mature database systems offer optimizations and caching. Application developers are proficient in SQL (Structured Query Language) and database models. SQL provides an easy way to analyze and aggregate the data and then to *massage* the data into the form needed by the client to create plots.

The BAJer is using PostgreSQL[2] because it is first and foremost open source and freely available. Secondly it is a mature database with a big feature set that is widely used. Should the need arise to re architecture the system in the future a good database can help a lot to adjust to the new needs.

I spent a lot of effort to model the relations between the data accordingly. As can be seen in fig. 2.2 on the next page the `papers` table is at the core. It does store the basic properties of a paper like the bibcode, title, abstract, the journal it was published in, the publication date and the first author. Around this table are several other tables and their corresponding junction tables. Junction tables are a concept to store many to many relations (n:m) between two other tables.

The `keywords` table and the associated `keywords_papers` junction table are an example of such a construct. Keywords are put into the first table and get their unique ID. The junction table stores pairs of keyword IDs and paper IDs (bibcode), thus mapping a relation between a keyword and a paper.

On the left side of the figure are the tables storing countries and authors. Additionally, to the connection via the junction table there is a connection directly to the papers table. These are to directly connect the first author and the first author's country to the values stored in the countries and authors table.

The table `citing_papers` stores all the citing papers. Several details are stored in their own columns to have easier access when accessing the data to do statistics on it. The `first_fetched` column stores at which point in time the citing paper became available in the database. This column is closely related to the last big part of the database model.

The last important part of this model can be seen on the right side of the figure: the

1 https://gitlab.com/thebajer
2 https://www.postgresql.org/

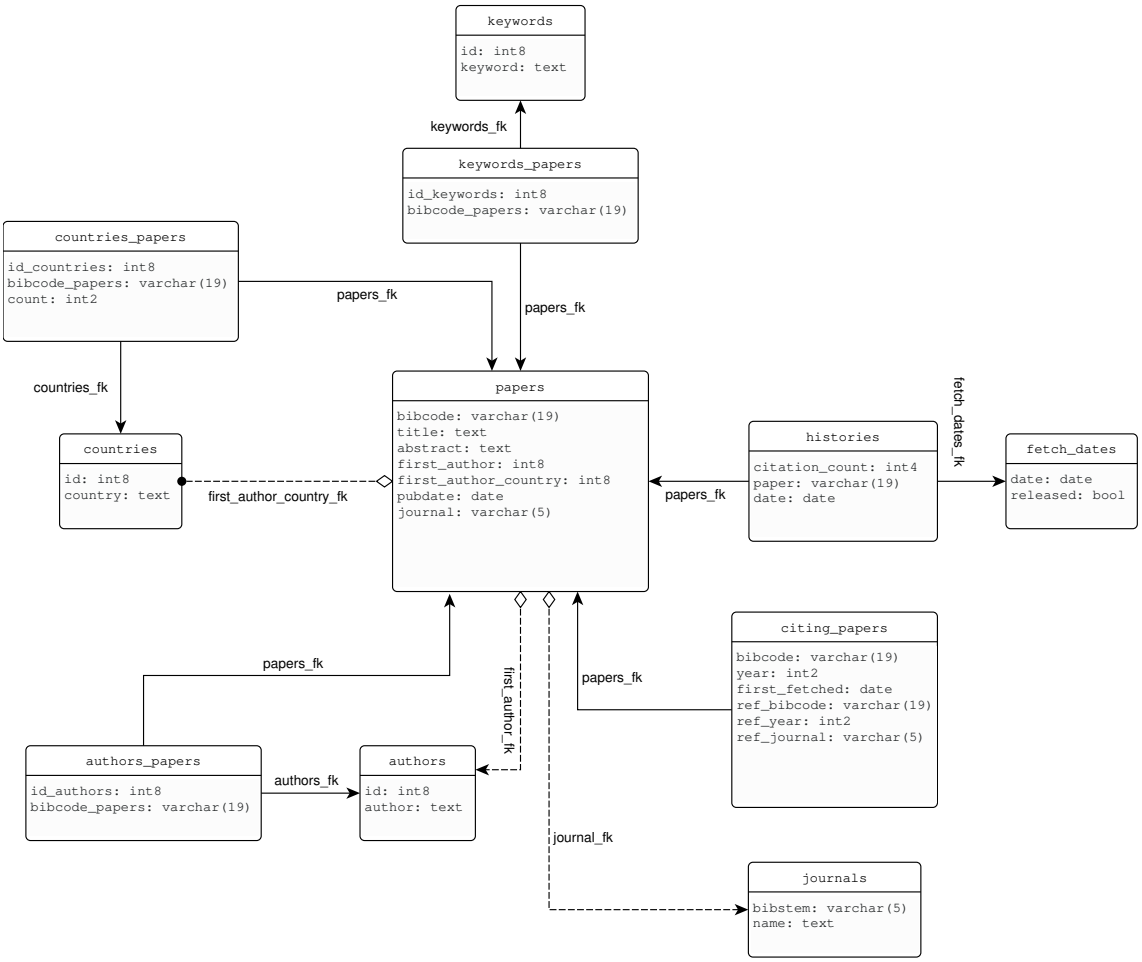


Figure 2.2: The database model of The BAJer

`histories` and `fetch_dates` tables. Each time the importer is run, which could be called "fetch run", a fetch date is inserted into the fetch_dates table. The `histories` table links papers to a fetch run and stores the changing data between imports. At the point of writing this it is only the citation count. A paper will, hopefully, have a higher citation count on another import that is run after some time has passed.

Read and write access is clearly separated as the importer is the only part of the system writing to the database and the API server is only reading from it.

2.5 Importer

2.5.1 Code statistics

The following table gives a glimpse at the size and complexity of the code for the importer.

Table 2.1: Detailed statistics on the code written for the importer of The BAJer

Language	File	Blank	Comment	Code
make	`Makefile`	1	0	4
Markdown	`README.md`	9	0	42
Python	`bajer-importer/countries.py`	3	5	305
	`bajer-importer/filters/author.py`	0	1	5
	`bajer-importer/filters/bad_words.py`	5	4	27
	`bajer-importer/filters/__init__.py`	0	0	1
	`bajer-importer/importer.py`	21	5	83
	`bajer-importer/__init__.py`	0	3	0
	`bajer-importer/__main__.py`	0	0	1
	`bajer-importer/runtests.py`	5	1	15
	`bajer-importer/store.py`	51	197	264
	`tests/filters/test_author.py`	6	0	14
	`tests/filters/test_bad_words.py`	6	0	12
SQL	`bajer-importer/bajer.sql`	153	155	145
		260	371	918

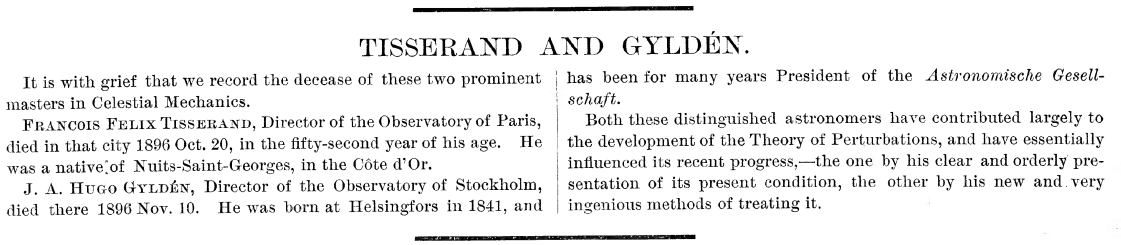
2.5.2 Description

The importer is written in the python programming language[1]. To access the ADS API[2] easily and without reinventing the wheel a python module created by Vladimir Sudilovsky & Andy Casey[3] is used.

The following fields are fetched for each paper: `title`, `pubdate`, `abstract`, `author`, `bibcode`, `citation_count`, `citation`, `date`, `first_author`, `indexstamp`, `keyword`, `year`, `doi`, `aff` and `pub`.

There are some types of papers tracked by the ADS that are irrelevant in regard to bibliometric statistics. For example announcements, proceeding reports, notices or even obituaries. The whole range of errata, addenda and the likes is also something that is not relevant and should be filtered out.

1 https://www.python.org
2 https://github.com/adsabs/adsabs-dev-api
3 https://github.com/andycasey/ads



TISSERAND AND GYLDÉN.

It is with grief that we record the decease of these two prominent masters in Celestial Mechanics.

FRANCOIS FELIX TISSERAND, Director of the Observatory of Paris, died in that city 1896 Oct. 20, in the fifty-second year of his age. He was a native of Nuits-Saint-Georges, in the Côte d'Or.

J. A. HUGO GYLDÉN, Director of the Observatory of Stockholm, died there 1896 Nov. 10. He was born at Helsingfors in 1841, and has been for many years President of the *Astronomische Gesellschaft*.

Both these distinguished astronomers have contributed largely to the development of the Theory of Perturbations, and have essentially influenced its recent progress,—the one by his clear and orderly presentation of its present condition, the other by his new and very ingenious methods of treating it.

Figure 2.3: Example of an obituary from 1896. Bibcode 1896AJ.....17...32.

To filter those the importer runs a set of checks on each paper. As soon as a check returns a negative result the import of the paper in question is aborted. The reason as to why the paper was discarded is written to the importers log file. Once a paper passed all checks it is stored in the database.

```
2019-05-15 09:33:43,493 - INFO - bajer_importer.runtests - Filtered 2012AJ
....143...76G because of: Found bad word --erratum-- in title.
```

Listing 2.1: A sample logline for a filtered paper.

2.5.3 Checks

A check is a python script which has a defined function that takes one paper as argument. If the check is passed it will return true. In case the check is not passed it returns a value of false along with the reason why the check failed, the returned reason is then written to the importers log file. A good example of how short a check can be is shown in listing listing 2.2 on the facing page.

The first check performed on each paper is a list of so called *bad words* that is matched against the title and keywords. If it encounters any the paper will be discarded. The *bad words* at the time of writing this thesis are: `errata`, `erratum`, `addendum`, `letter`[1], `supplement`, `corrigendum`, `proceedings`, `announcement`, `editoral`, `obituary` and `notice`. This test should filter out many papers that are not relevant to bibliometric analysis.

The second check discards any paper that does not have a first author. The downside of this check is that it will also generate a considerable amount of false positives with old data. This caveat is accepted to have better results with current papers that are of higher interest.

The bibcode of the ADS is used as *unique key* for the papers. This should eliminate any possibility to have duplicate papers unless the same paper got assigned two different bibcodes.

A more detailed discussion of an import and discarded papers can be found in section 2.8 on page 16.

1 Publications with the word *letter* in the title or keywords usually are not research papers but notices and announcements. Letters in the scientific sense usually doo not contain the word *letter* in the title or keywords.

```
def dofilter(row):
    """Filter for empty first_author"""
    if row.first_author is not None and row.first_author.strip():
        return (True, None)
    else:
        return (False, "First author empty.")
```

Listing 2.2: Sourcecode of the check if the first author is empty.

2.5.4 Extracting Country Information

Extracting the countries of the authors from the ADS data proved to be difficult for a number of reasons. The first and most obvious one is that there is no dedicated field for this. The one field that does carry some country information is the affiliation (`aff`) field. If available there is one affiliation line for each author of the paper. Unfortunately it does not have a well-defined format and the quality of present country information varies. In listing 2.3 is a sample showing the major differences. Countries can be either written out fully, abbreviated like USA, UK or not be present at all.

Another common occurrence are papers from the United States that completely omit a country but luckily include the two-letter code of the state which can be matched.

```
Department of Astronomy, California Institute of Technology, MS 105-24, Pasadena, CA 91125, USA
Centre for Astrophysics and Supercomputing, Swinburne University, Hawthorn, VIC 3122, Australia
National Astronomical Observatory of Japan, 2-21-1 Osawa, Mitaka, Tokyo 181-8588, Japan.
Steward Observatory, University of Arizona, 933 North Cherry Avenue, Tucson, AZ 85721
School of Physics and Astronomy, University of Birmingham, Edgbaston, Birmingham B15 2TT;
```

Listing 2.3: A sample showing the difference in affiliation data.

The last issue with a free text field are typographic errors. How those errors get there is not clear. Whatever the cause is, typographic errors were detected during the implementation of the importer and need to be dealt with.

A good method to detect them is to use the Levenshtein distance (Levenshtein, 1966). It is a metric of how many steps need to be taken to transform one word to another. These steps can be one of the following: changing, inserting or deleting a character. The python module `python-Levenshtein`[1] is used for this step.

The method to extract a country from an affiliation needs a list of countries to test against. Using the `pycountry`[2] module a list of countries was generated. The list was manually enhanced if an abbreviation or a different way of writing a country is common. A list of two letter abbreviations for US states has been added as well.

The method to extract the country from an affiliation line works as follows:

1 https://github.com/ztane/python-Levenshtein
2 https://pypi.org/project/pycountry/

1. First the whole line is converted to lower case character for easier comparison. Accents are removed and HTML entities, which sneak in sometimes, are converted back to the actual characters.
2. The list of countries is matched against full words with a regular expression.
3. If no match has been found, the line is then split into single words and separator symbols like commas and semicolons are stripped.
4. The next step is to apply the Levenshtein distance and see how well the words match. The `ratio` method of the `python-Levenshtein` module is used. It will return a floating point number between 0 and 1 with 1 being a perfect match and 0 if no letters match. In order to avoid false positives between two countries the threshold has been carefully selected. A test has been done by calculating the ratios of all countries in the list against each other. The highest ratio is between *Austria* and *Australia* with 0.875. Therefore, the ratio needed to be considered a match is set to 0.88.
5. If still no country has been identified at this point, a last test is performed to check if any of the US states is present. In order to avoid false positives, the original affiliation line is used as the two letter abbreviation for a US state is usually written in capital characters.

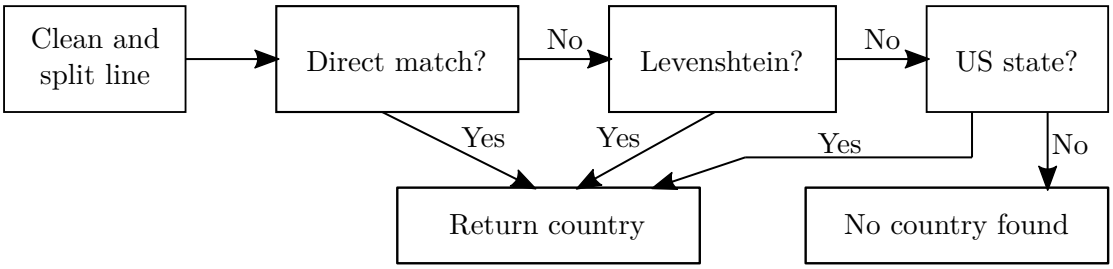


Figure 2.4: The flow diagram of the steps involved to extract a country from the affiliation field.

With this approach the number of false positives should be almost zero and countries should be identified reliably.

2.5.5 Storing

At this point during the import the information of the current paper is written to the appropriate tables of the database and the relations between the different properties are created.

The bibcode encodes information like the publishing year and the journal in a 19 character long string (*ADS Bibcode* 2019). This makes it very easy to extract data from it by slicing the string.

The publication date received from the ADS is not a valid date format. It does report the year but not always the month and never the day. Omitted data is replaced by zeros. For example `2019-04-00`. To have a valid date that can be stored in the database `-00` is replaced with `-01`.

If there was no country found for the first author but for one of the other authors, the first found country will be stored as the *first authors country* to have at least some country information on the paper.

2.6 API Server

2.6.1 Code statistics

The following table gives a glimpse at the size and complexity of the code for the API server.

Table 2.2: Detailed statistics on the code written for the backend of The BAJer

Language	File	Blank	Comment	Code
Bourne Shell	`freebsd-rc-file`	8	10	22
Markdown	`README.md`	10	0	23
Python	`app.py`	14	0	27
	`cache/__init__.py`	6	4	17
	`controller.py`	49	270	163
	`models.py`	33	1	60
	`orm.py`	3	0	12
YAML	`swagger/swagger.yaml`	0	0	587
		123	285	911

2.6.2 Description

The API (Application Programming Interface) server is written in python and offers a REST API. In order to have a well-defined API the OpenAPI specification in version 2.0 is used(*OpenAPI Specification (v.2.0)* 2019). With this specification the API endpoints, which parameters they accept, what the answer looks like and which function needs to be called can be defined in a single file.

The benefit is easier testing (not implemented at the time of writing this thesis) and easier integration as there are tools and libraries that can create the necessary code from such a file. The connexion module[1] is used. It handles the server side, creates the API endpoints according to the specification and does input checking and sanitation. This speeds up the development process a lot and the only thing left to implement are the actual functions that serve the API endpoints.

As can be seen in listing 2.4 on the next page the actual API functionality to implement is the SQL query and turning the results of the database into a JSON object that is then returned to the client. Many of the plots require data in the same format: bibstem, x-axis, y-axis. This common data format helps as many steps in running the query and converting the database result into a JSON response can be abstracted and used by multiple function.

1 https://github.com/zalando/connexion

Extra time is needed for more unusual plots that need the data in a different format. The steps to convert the query results into a JSON object need to be implemented specifically for their requirements.

```
def bibstem_get() -> str:
    res = db_session.execute(
        """SELECT bibstem, name
        FROM journals
        ORDER by bibstem"""
    )
    results = {}
    for row in res.fetchall():
        results[row[0]] = row[1]
    return json.dumps({"bibstem": results})
```

Listing 2.4: Sourcecode of the `/bibstem` API endoint.

The following listing (listing 2.5) shows a more elaborate but still standard implementation of an endpoint.

```
def published_get(fetchDate, minYear, maxYear, bibstem) -> str:
    db_query = """SELECT
    papers.journal as bibstem,
    extract('year' from date_trunc('year',papers.pubdate)) as year,
    count(histories.paper) as paper_count
FROM
    papers
JOIN
    histories
ON
    histories.paper = papers.bibcode
WHERE
    histories.date =  :fetchDate
AND
    papers.pubdate >= :startYear
AND
    papers.pubdate < :endYear
AND
    papers.journal IN :journals
GROUP BY
    year,
    bibstem
ORDER BY
    bibstem,
    year;"""

    return run_2d_query(
        fetchDate, minYear, maxYear, bibstem, db_session, db_query
    )
```

Listing 2.5: Sourcecode of the `/published` API endpoint.

To speed up the development by reducing the load on the development computer a simple caching mechanism was implemented. The database query with the necessary parameters is hashed and that hash is stored alongside the result in a python dict. On each subsequent API request the hash is generated again and if it is stored in the cache the already stored result is taken. This cache resides fully in running memory. It will be either removed, disabled or replaced by a better solution when put into production as it is prone to a denial of service attack in the current state.

A potential attacker can request many slightly different requests. For example by iterating over the start and end years. Besides increasing the server load this will also blow up the cache, potentially to the point where the machine will be running out of memory.

2.7 Client

2.7.1 Code statistics

The following table gives a glimpse at the size and complexity of the code for the web-based client.

Table 2.3: Detailed statistics on the code written for the frontend of The BAJer

Language	Files	Blank	Comment	Code
HTML	1	0	1	20
Markdown	1	11	0	31
SVG	1	0	0	4
JavaScript	24	61	90	446
Vuejs Component	26	177	150	2715
	53	249	241	3216

2.7.2 Description

A web based client is the best option to enable a wide audience to interact with the data. The options for a modern JavaScript framework for full applications were React[1], Vue.js[2] and Angular[3].

After evaluating all three candidates the choices has been made in favor for Vue.js. With its great documentation and reliance on standard HTML[4] and CSS[5] for its components it seemed to be the best choice for someone that has not done any modern web application development. The Vuetify[6] component library was used to speed up the development. It offers a wide variety of GUI[7] components, thus making it very easy to create a modern

1 https://reactjs.org/
2 https://vuejs.org/
3 https://angular.io/
4 Hypertext Markup Language
5 Cascading Style Sheets, the part that makes HTML look good
6 https://vuetifyjs.com/
7 Graphical User Interface

and functional interface.

As the functionality of the client revolves around the graphing of any data the API server provides a good plotting library is needed. Most JavaScript based graphing and chart libraries are targeted towards business applications and lack more in depth features that a scientific audience is used to. With Plotly.js[1] a very feature complete library was found nonetheless.

2.7.3 Main Screen

The main screen of the client features the dashboard containing every plot that is created. Each plot is embedded in its own container element. This container can be resized and dragged to create a customized dashboard. At the top is the title bar with the name giving title on the left side. The title can be edited right there by clicking on the edit / pencil icon. On the right side in the title bar is a set of tabs which allow switching between the settings that define the plot, the optional description and the plot itself. The container element is finished at the bottom by the status bar. Once a plot has been created does show the chosen settings in a compact form.

When creating a new plot the *SETTINGS* tab is active to show the form as can be seen in the top right part of fig. 2.5 on the next page. First is the field to select which fetch date to use. The user can choose which journals are to be considered when creating the plot. At the core of the interface to select the years of interest is a slider. Having close to 200 years to choose from makes a slider a somewhat difficult component to accurately select the years as the step size gets too small. To remedy this, the option to limit the starting point of the slider has been introduced. It offers the very first year with data available, 1827, and then continues in steps of 50 years. The default is set to 2000.

Finally, the last field lets the user select what kind of data shall be visualized in a plot. Next to it is the button that will start the creation of the new plot. Once clicked the active tab for this plot is changed to the *PLOT* tab. The data is retrieved from the API server and a loading animation is shown for as long as this takes. As soon as the data has been received it is handed over to plotly.js which then creates the graph.

The top left of fig. 2.5 on the facing page shows such a graph. Beneath the title bar is a set of buttons and other elements which offer actions that are specific to this graph. The actual plot does take up the most area. When hovering the mouse over it plotly.js shows its own toolbar which contains several tools to zoom and pan within the plot. Out of the box it offers to download the shown plot as a PNG image. As a pixel based image has limitations, the configuration of plotly.js has been modified to enable the download as a vector based SVG image as well.

The toolbar below the title bar offers actions that directly concern the graph itself. Most commonly available are two options to download the data as either CSV[2] or JSON[3]. The possibility to toggle the visibility of the legend and the title within the plotly.js environment help to adjust usable space and information present when saving the graph as an image.

1 https://plot.ly/javascript/
2 comma separated values
3 JavaScript Object Notation

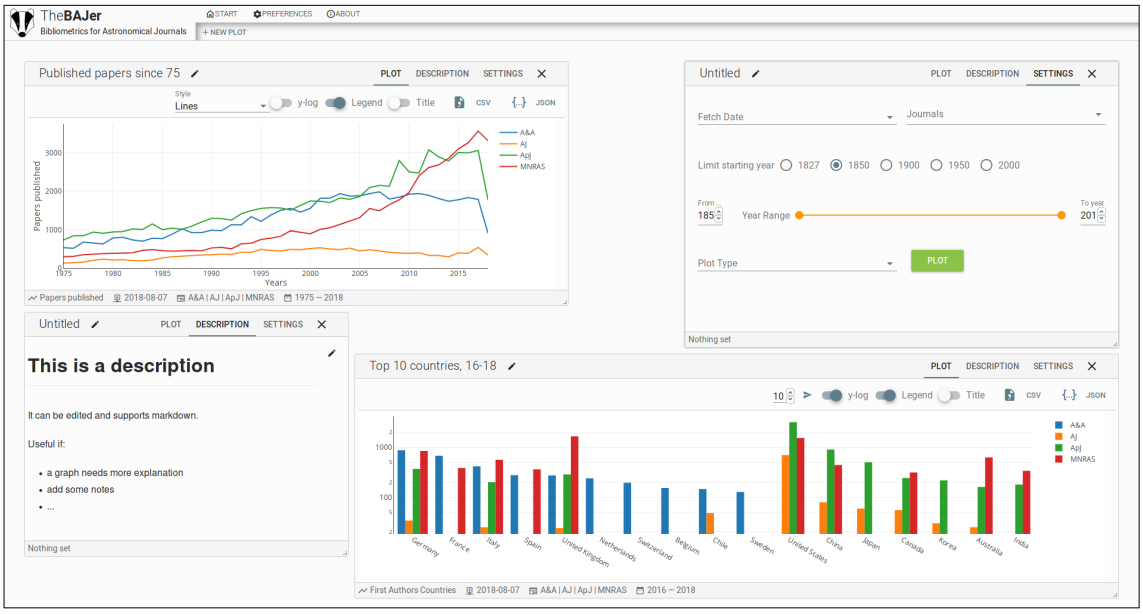


Figure 2.5: Example of the main screen with a customized dashboard.

Setting the y-axis to a logarithmic scale is also possible for most plots. The user can choose between several styles to find the best fitting for the current plot.

At last there is the *DESCRIPTION* tab. Here the user can add text to either add more information or take notes. Markdown syntax is supported to structure the text. The future will show if this feature is useful.

2.7.4 Preferences

The feature to change the colors of the journals in the plots is located at the preferences screen. Another big part is the session / application state management. Here the user can reset the whole application and export the current session. The session can be saved to a file or copied into the clipboard. When a session is imported the user can choose if it should replace the current session or if it should be merged. Sessions are discussed in more detail in the following section.

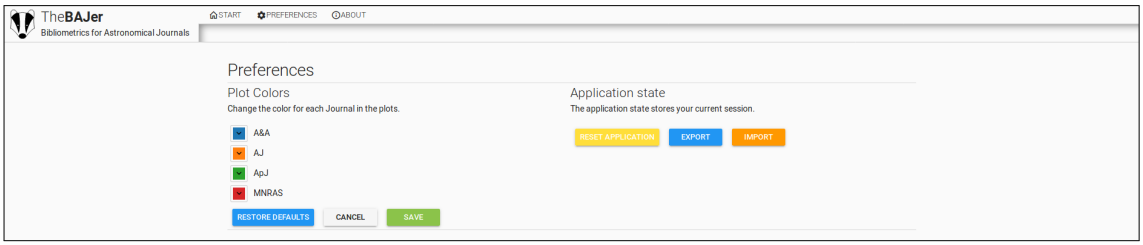


Figure 2.6: The preferences screen.

2.7.5 Technical Details

The settings how many plots have been created with which settings and other preferences like the color settings are considered the application state. It is stored in serialized form in the web browsers local storage. On opening the website or on reloading the stored state is being inflated. The data of plots is not stored this way and needs to be retrieved from the API server.

In order to be able to merge different application states the plots are identified not by an incremented integer number but by an UUID[1]. The chances of creating two UUIDs of the same value are practically zero.

Currently, the application state is stored in one defined location in the browser's local storage. There is no technical obstacle to store more than one state in the user's browser. A more sophisticated session management system that allows the user to quickly switch between several stored sessions is a feature for the future that could be implemented with reasonable effort.

2.8 Analysis of an Import

An exemplary import has been performed on May 15th, 2019. It started at 09:32am CET and ran until 11:28am. During that 260 217 papers were retrieved from the ADS and processed.

2.8.1 Processed papers

As can be seen in table 2.4 around 5 % of papers are discarded during the import. A more detailed overview as of why a paper has been discarded is provided by table 2.5 on page 19.

Table 2.4: Statistics on papers filtered during import

Papers found	260 217	100.00 %
Papers filtered	14 208	5.46 %

To put the following discussion into perspective a look at the number of papers imported, and thus the amount of papers published, is necessary. Figure 2.7 on the next page shows the increase in papers published starting after World War 2 in the 1950s and a sharp increase starting in the early 1970s.

Figure 2.8 on the facing page shows how A&A started publishing at about the time when the output of the observed journals started to increase. While ApJ and even more so MNRAS kept increasing the yearly output, A&A's output leveled out since the early 2000s. AJ followed the trend but with a much shallower increase. Like A&A its output started to decline in the early 2000s, but it has been increasing in recent years. The sharp decline on the right end is due to the import being run in May with half a year to publish left.

The gap in the data for the Astronomical Journal (AJ) is because it ceased publication with the start of the American Civil War in 1861 and managed to publish the next edition in 1886 (Gould, 1886).

1 Universally unique identifier – https://en.wikipedia.org/wiki/Universally_unique_identifier

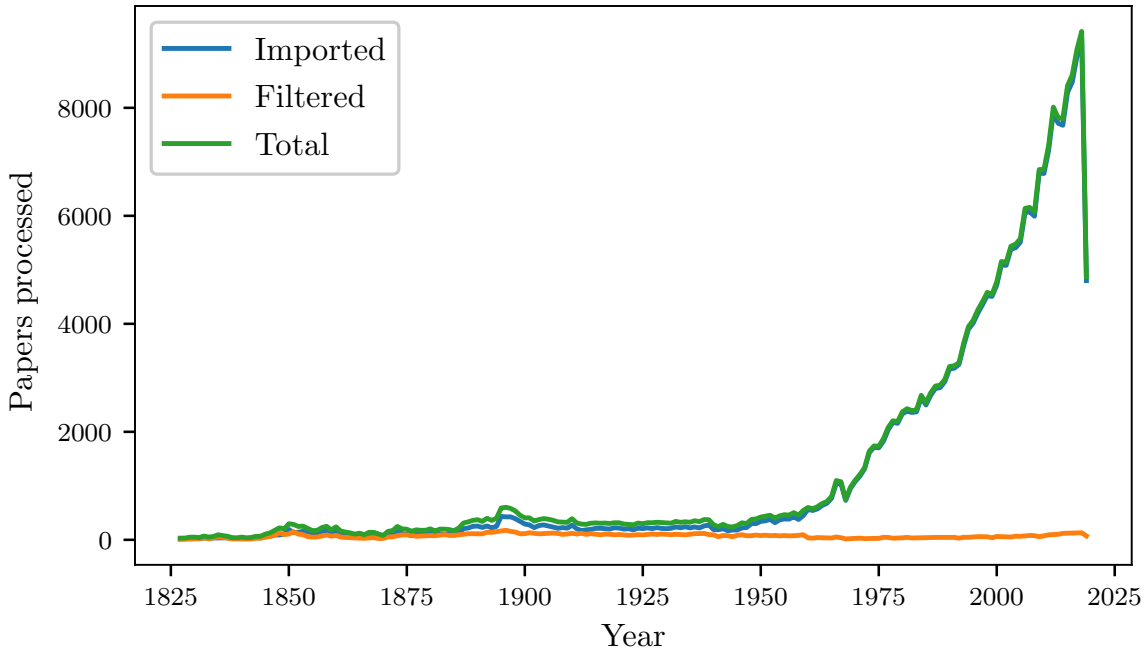


Figure 2.7: The number of papers processed.

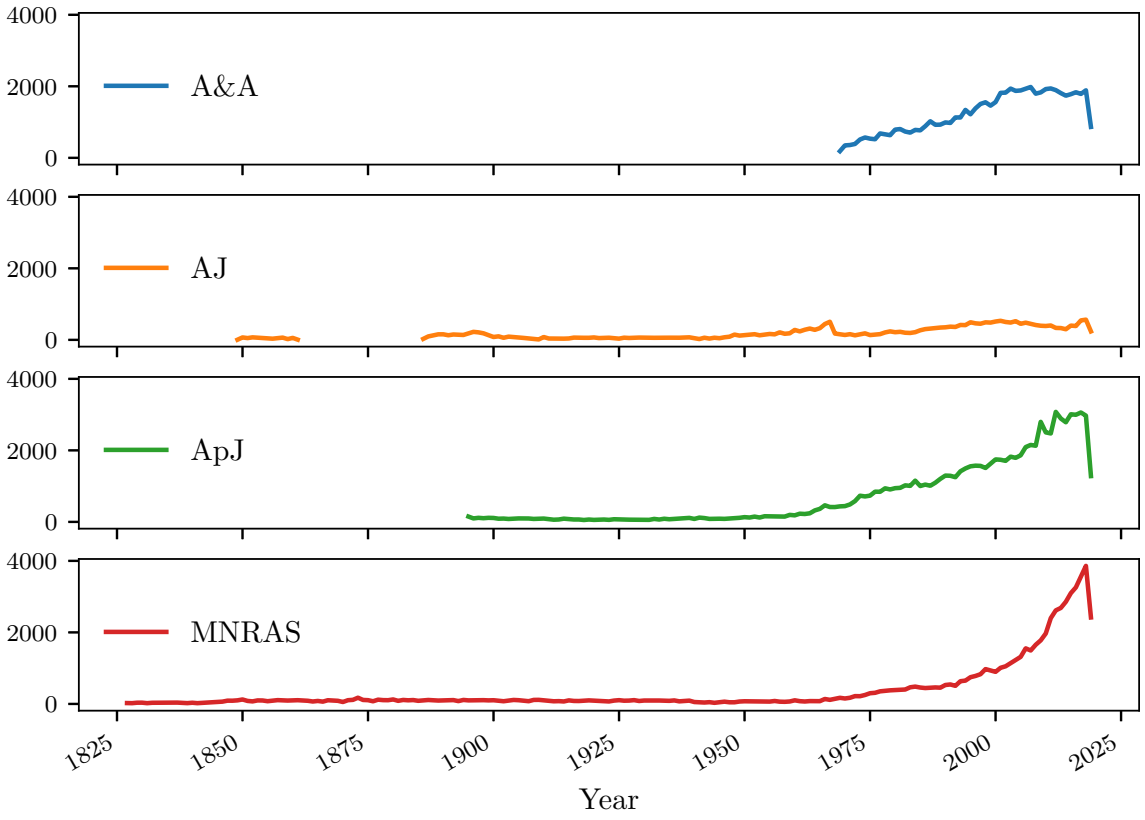


Figure 2.8: The number of papers imported.

Figure 2.9 shows the absolute number of papers filtered out over time. The increase in the last decades can be attributed to the higher number of papers published overall and is predominantly evident in the journals with a high paper output like MNRAS or ApJ. In fig. 2.10 the same data is shown normalized to the number of papers published. It can be seen that for the last decades the amount of papers filtered is in the low single digit percentage.

One thing that is very apparent in fig. 2.9 and fig. 2.10 is the large amount of papers filtered up until around 1960. One possible cause is the steep decline in *obituaries* and *proceedings*, see fig. 2.12 on page 20. The other cause is the bad quality of the data manifested in the missing first authors. Figure 2.11 shows that the majority of papers without a first author falls into the same time frame. This combined with the overall lower number of papers published during this period results in the rate of filtered papers going up to around 50 %.

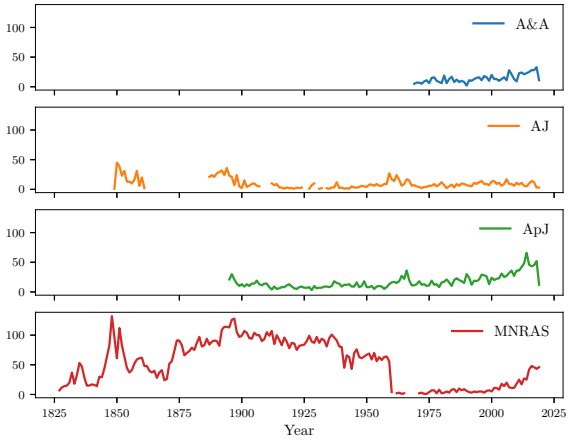


Figure 2.9: Absolute number of papers that have been filtered during the import.

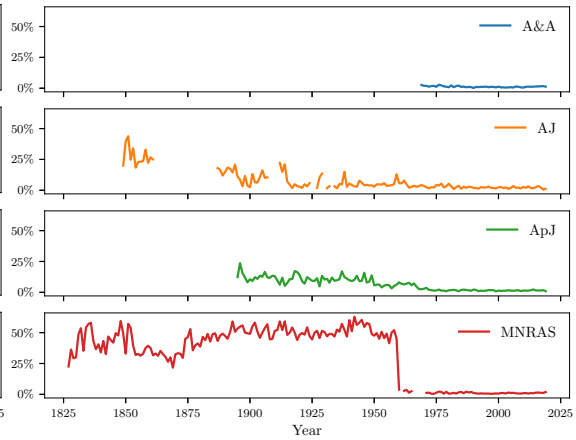


Figure 2.10: Normalized number of papers that have been filtered during the import.

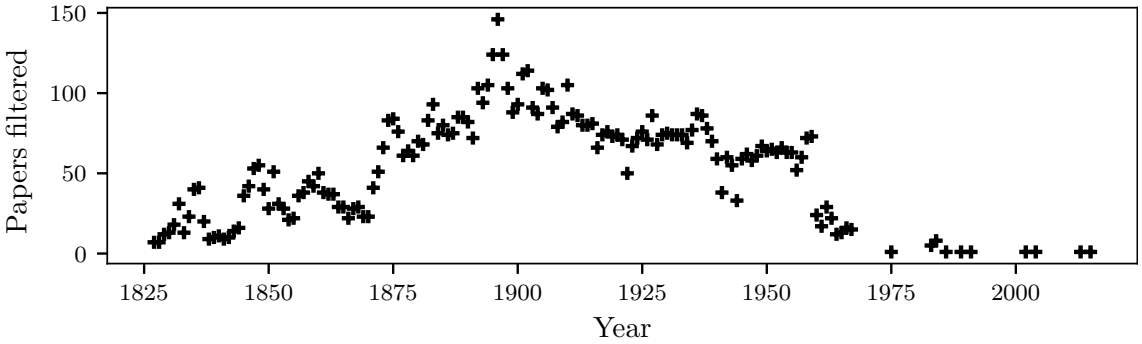


Figure 2.11: Absolute number of papers discarded because they do not have a first author.

2.8.2 Details on filtered papers

The way the filter works during the import needs to be considered when looking at all the statistics that follow in this chapter. As discussed in section 2.5 on page 7 the first check

is looking for bad words, first in the title and then in the keywords. It will discard the currently checked paper at the first found *bad word*.

Therefore, a search for a specific keyword in the ADS can very well result in a higher number of papers found. Out of the 5971 papers that were discarded because a *bad word* has been found, the majority had that word in the title. A total of 5884 papers had it in the title while only 87 were discarded because one of the keywords was on the list of *bad words*.

To avoid any misunderstandings, it should be noted that the word *Letter* does not refer to the modern kind of scientific letters used in journals. The modern kind of letters do not contain the word *letter* in the title or keywords. The word *letter* fell out of mainstream use in the early 20th century as can be seen in fig. 2.12 on the following page. In the early times of journals, letters were used to discuss organizational issues, for corrections, telling other astronomers how well a certain new telescope worked and to a certain extent to communicate observations made.

Table 2.5: Detailed statistics on why papers were discarded during import

	A&A	AJ	ApJ	MNRAS
No first author	16	365	618	7231
Bad words found	695	986	1478	2812
Errata	59	194	89	291
Erratum	392	512	1198	607
Addenda	0	1	0	0
Addendum	4	21	27	18
Letter	8	175	63	203
Supplement	6	19	7	20
Corrigendum	212	0	4	7
Announcement	3	7	22	52
Editoral	11	17	37	10
Obituary	0	17	4	1149
Notice	0	23	27	455

The second check every paper has to pass is if there is a first author present. When looking at the statistics it becomes apparent that the older journals have an expected higher number of papers filtered. This is due to the fact that older data tends to be worse in quality. For example, it is missing the first author.

With all this data present the question arises how the use of certain words changed over time. The filtered words in titles have been aggregated with the mean value over 10 year ranges for better readability. A&A switched in the early 2000s from using *erratum* to *corrigendum* as can be seen in fig. 2.12 on the next page. The other journals started to use

erratum exclusively in the past decades from a mix of *errata*, *erratum* and the occasional *corrigendum*. MNRAS used to have *obituaries* and *proceedings* from the mid 19th century to the late mid 20th century.

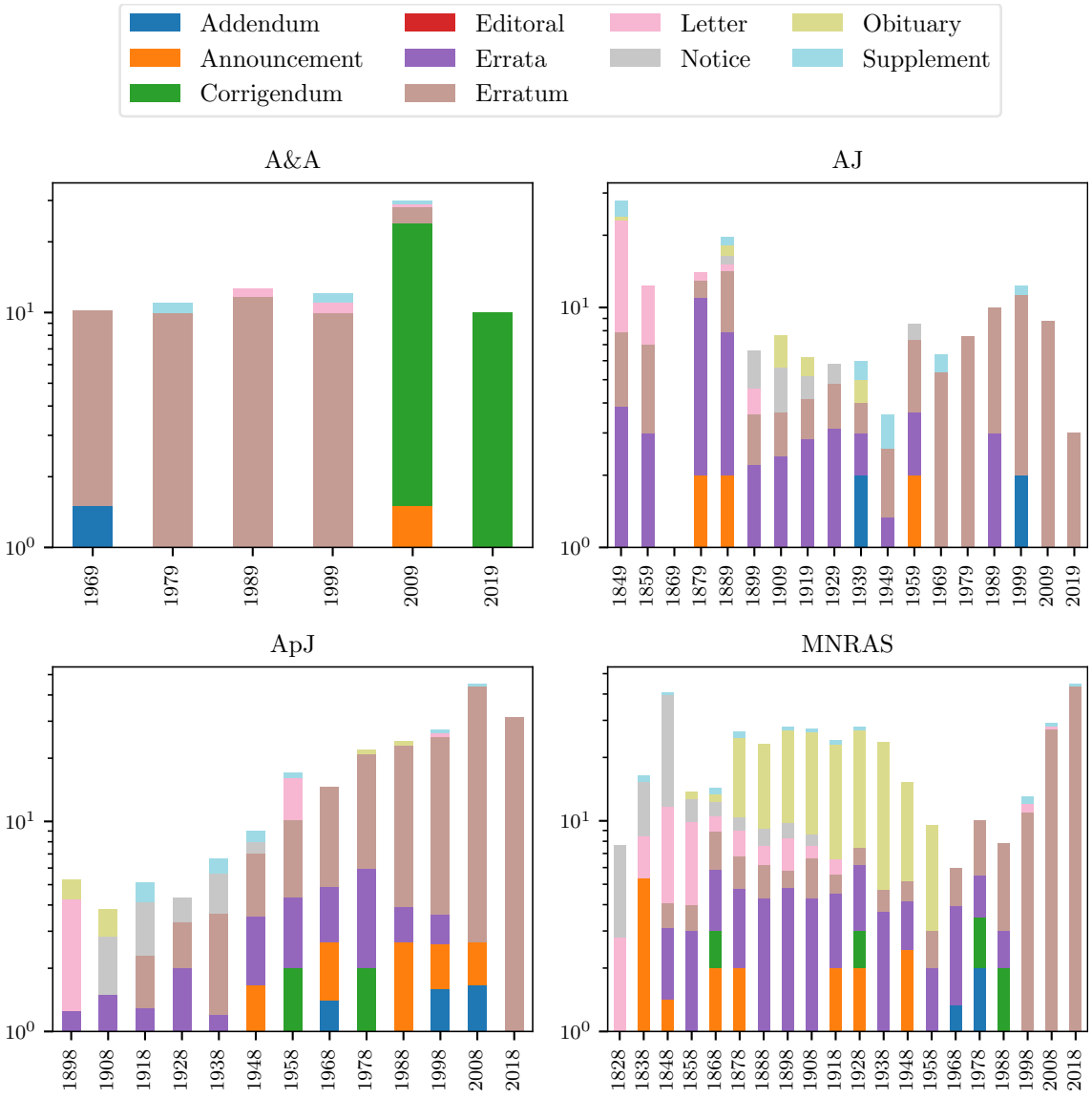


Figure 2.12: The use of words like errata, erratum, etc. in titles. The data has been binned into ranges of 10 years for better readability. Bins are labeled on the left side. Letters that are filtered out are not the modern type of scientific letters. In the 19th and early 20th century, they were used to discuss organizational issues, for corrections and to a certain extent to communicate observations made."

2.8.3 Country Data

As discussed in more detail in section 2.5 on page 7, extracting the country information from the affiliation is not straight forward. With the log output of the importer it is possible to visualize the amount papers for which no country has been found. The following plots 2.13 and especially 2.14 show that for the most time there is not enough good data available. Almost complete and therefore usable data is available only since about the year 2000, except for MNRAS for which the situation only got better in the last few years.

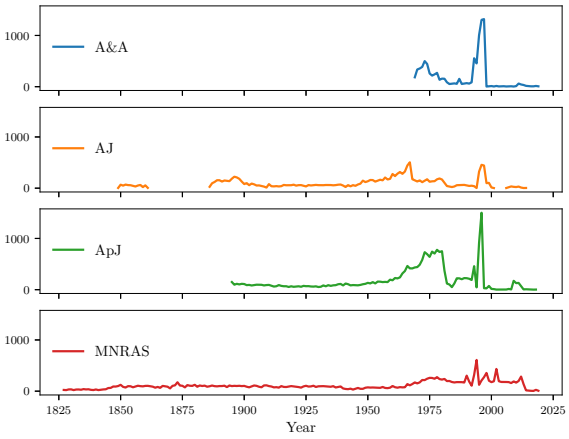


Figure 2.13: Absolute number of papers with no country data available.

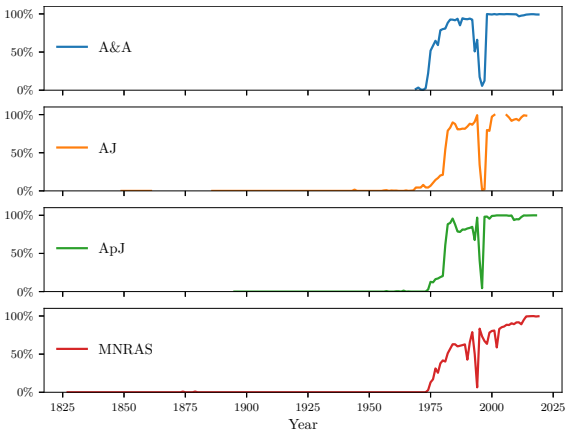


Figure 2.14: Normalized view of papers with country data available.

The low number of papers with country data available in the 1990s is clearly visible. For A&A, AJ and ApJ the year in which there is almost no country data available is 1996. MNRAS has the same situation in 1994. There is almost no affiliation data available in the ADS for these years. The cause for this peculiarity is not the topic of this thesis and will most likely need insight into the procedures of the ADS for the affected years.

To validate that the extraction of country data works sufficiently well, I selected 100 random papers from the available data and validated the country stored for the first author. Out of the 100 papers, 86 papers were a correct match, 10 were a partial match which matched one of countries listed, and 4 matched the country of an author other than the first author as there was no country found for the first author. A table with the randomly selected papers and the individual results can be seen in appendix A.1 on page 41. Figure 2.15 shows the final result.

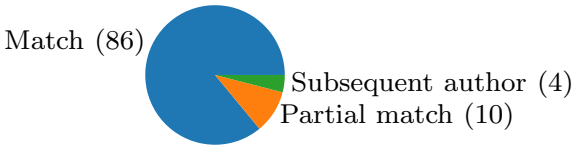


Figure 2.15: Results of verifying the country extracted from 100 randomly selected papers.

The importer in its current form can introduce some bias as it searches the list of countries in the affiliation. Countries earlier in the alphabet will win over countries starting with a letter further back.

CHAPTER 3

Results

3.1 Graphical results from The BAJer

The following plots were created by The BAJer, saved as SVG and converted to EPS to be included in this thesis. The time frame was selected to include all available data, starting in 1827 and ending in 2019. One has to remember that the quality of the data, and thus the ratio of papers filtered out, is not good further back in time as shown in Figure 2.10. The journals used in the following plots are "Astronomy and Astrophysics" (A&A), "The Astronomical Journal" (AJ), "The Astrophysical Journal" (ApJ), and the "Monthly Notices of the Royal Astronomical Society" (MNRAS).

Some artifacts are present in the plots because poltly.js does not handle the lack of data for the Astronomical Journal well which ceased publication between 1861 and 1886 (Gould, 1886).

3.1.1 Published

This plot shows the absolute number of papers published in each year.

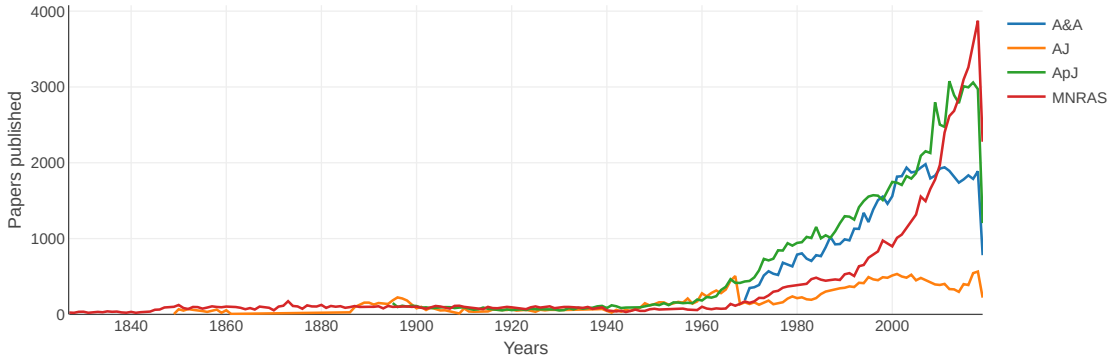


Figure 3.1: Published papers

The drop in the last year is well visible and expected because the data of the plot was fetched in the middle of the year when not all papers have yet been published. The significant rise in the amount of papers published can be seen to start in the 1950s and 1960s. Exponential growth is visible for the Monthly Notices of the Royal Astronomical Society (MNRAS). It overtook A&A and ApJ in 2014. The Astrophysical Journal (ApJ) managed to grow at a steady rate until the late 2000s when it managed to increase its output significantly, only to flatten out in the 2010s. Astronomy & Astrophysics (A&A)

kept pace with ApJ until the 2000s, when its output started to stagnate. The Astronomical Journal (AJ) does not seem to follow any trends. It has been growing slowly until the early 2000s, declined in the following years and started to grow fast in the past couple years.

3.1.2 Citations

This plot shows the absolute number of citations a journal gathered. The citations shown for a year are not the ones gathered in that year but for the papers published in that year.

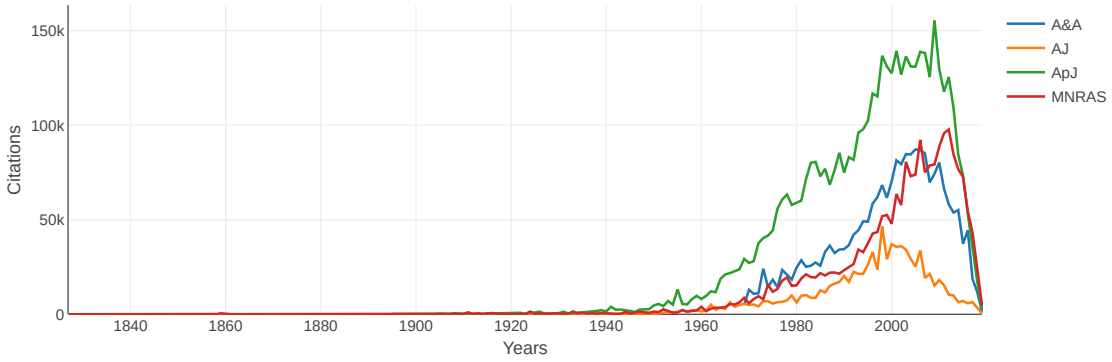


Figure 3.2: Citations

It can be seen how the citations grow with the increased output of the journals. The drop towards the recent years is starting earlier than in the plot of published papers. This is due to the fact that newly published papers need some time to get citations, something that can be seen in all the plots that deal with citations. Interesting is the fact that the Astrophysical Journal (ApJ) is getting the majority of citations.

3.1.3 Citations Normalized

This plot shows the number of citations normalized to the number of papers published.

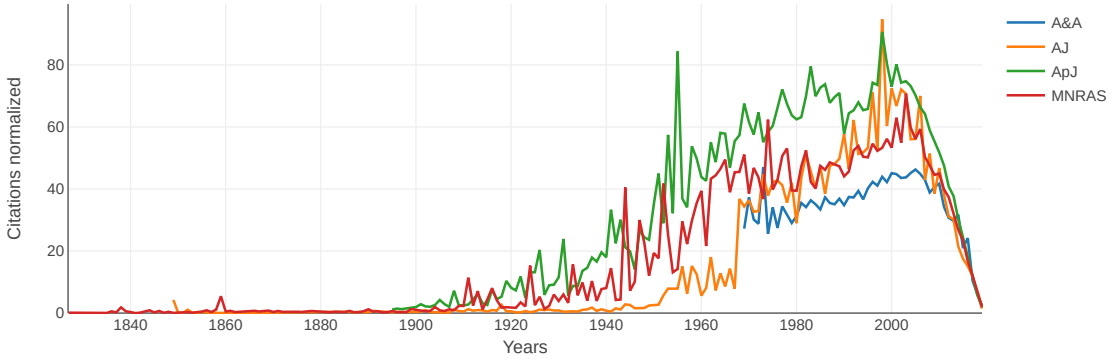


Figure 3.3: Citations normalized

The Astrophysical Journal (ApJ) is still dominating the citations, but its lead is reduced when compared to section 3.1.2.

The low, almost close to zero number of citations to papers published up until the early twentieth century is most likely caused by less academic interaction and incomplete data from that time in the ADS.

3.1.4 Journal Citations to all Citations

The following plot is showing the ratio of a journals citations to the overall citations of all journals included in the plot. Kurtz et al. (Michael J. Kurtz, 2018) showed this kind of plot in their blog series and called it *mind share*. It can be seen as a metric to measure the *mind* or *market* share of available citations. Another way to see it is as to how important a journal is seen as it can have a much higher share in citations than in published papers.

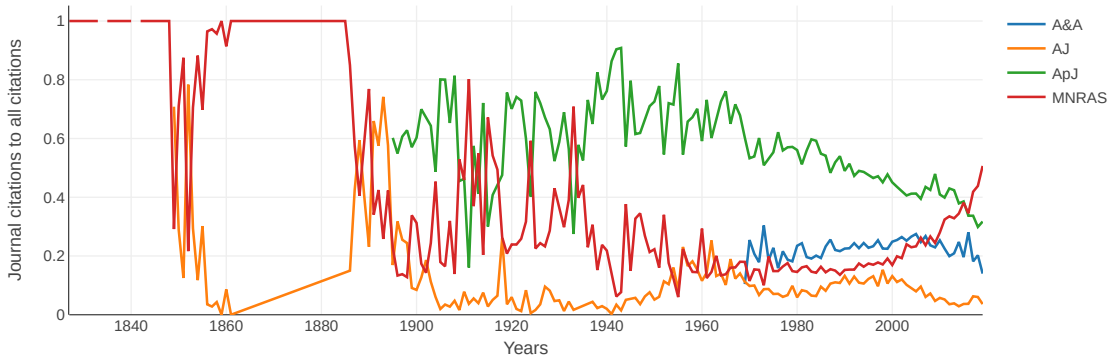


Figure 3.4: Journal citations to all citations

For this plot the data is getting interesting as soon as there is some competition. Starting in the early twentieth century ApJ and AJ start to publish next to MNRAS. A better visual representation is the same plot as a stacked line chart.

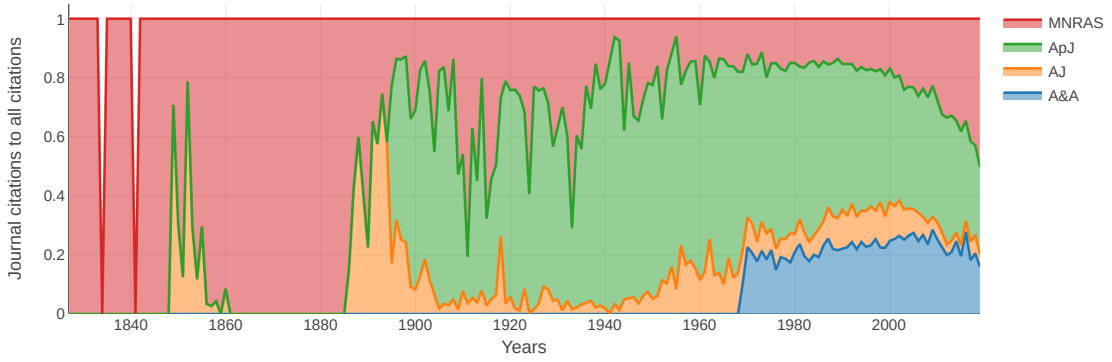


Figure 3.5: Journal citations to all citations as stacked line chart

3.1.5 Citations as box plot

This plot shows the citations of each journal per year as a box plot.

Figure 3.6: Citations as box plot

This is by far the most costly plot in terms of computing power for the client. For the other plots the data is aggregated by the database and only the data points needed to display the plot are transmitted to and stored on the client running in the user's browser, usually one data point per journal and year. This is not the case for this plot. All individual data points are needed in order for the plotting library to create the box plot. This results in much more data to be transferred from the server to the client and a much higher memory usage which can make the user's browser feel sluggish.

For this plot only the years from 2000 to 2019 are shown. This is for two reasons; first to not overload the browser with too much data and secondly, to still be decipherable as too many years would detract from the informative value. Papers with zero citations are omitted as the interest lies with papers which gathered any citations.

Even though the values of the boxes vary a bit from journal to journal it can be seen that the majority of papers stop gathering citations after about six to seven years.

Another feature that sets this plot apart. Outliers on the top can be clicked. When doing so, a new window will open the clicked paper in the web interface of the ADS.

3.1.6 Zero Citations

The following plot shows the absolute numbers of papers with zero citations. It can be seen that there are many papers with zero citations recorded in the ADS up until the 1960s to 1970s. The rise in recent years is expected as freshly published papers still need to gather citations.

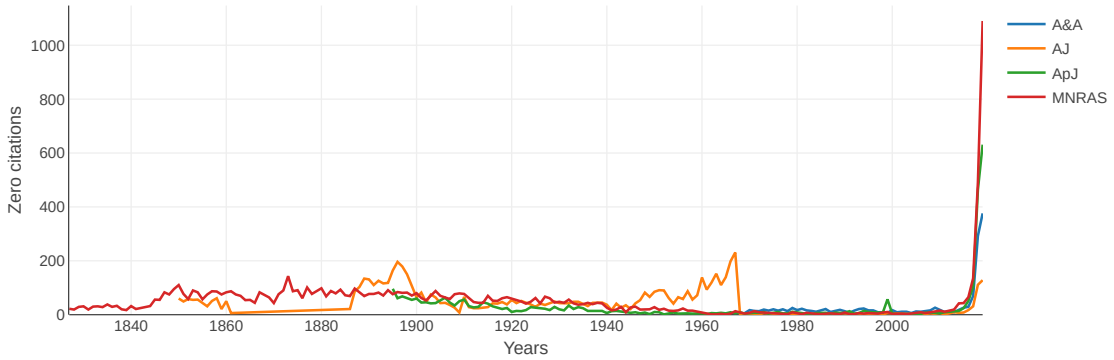


Figure 3.7: Zero citations

3.1.7 Zero Citations Normalized

This plot shows the number of papers with zero citations normalized to the number of papers published.

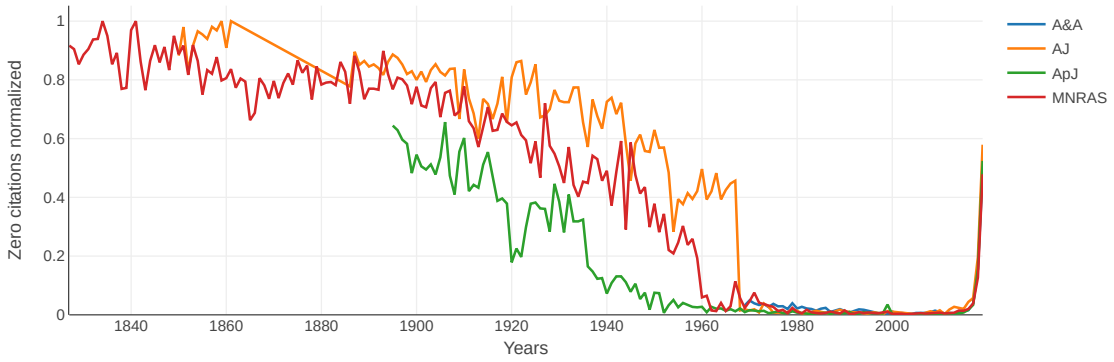


Figure 3.8: Zero citations normalized

The steep drops in the 1960s and around 1970 indicate that the data stored in the ADS got considerably better with papers published at that time. This should be considered when doing historic bibliometric research.

3.1.8 First Authors Countries

This plot shows the countries from which the most papers have been submitted in the selected time frame.

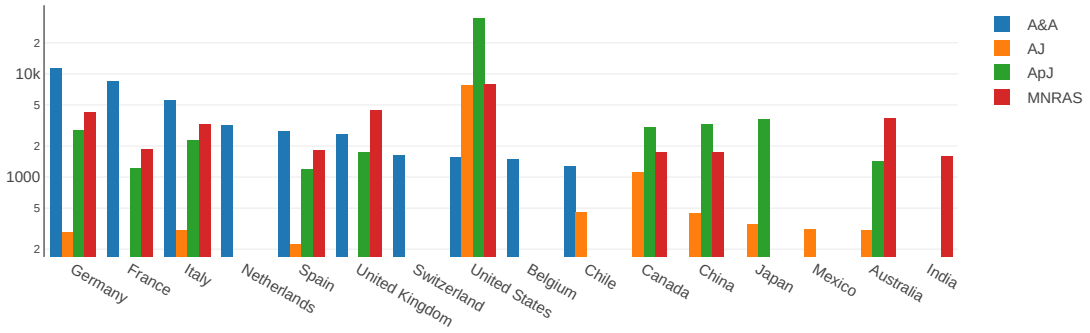


Figure 3.9: First authors countries, top 10 from 1829 to 2019

To interpret this data one should keep section 2.8.3 on page 21 in mind. The countries extracted are only good enough for the past several years. The order of the countries is based on the rank of the countries and the order of the journals. It can be more practical to create separate plots for each journal of interest.

CHAPTER 4

Conclusions

I created a software package that imports, cleans, and visualizes data from the ADS database. Country data, without a dedicated field in the database, is inferred from the affiliation data. The process to extract country data for the authors is a multi-layered process that takes typographic errors into account and the typical absence of country data when affiliated in the United States. The imported data is stored in a relational database. An API server provides the aggregated data to clients who then can visualize it. A web-based client that works in any modern web-browser is provided for easy interaction with the data. It enables the user to create plots and inspect the visualized data interactively. During an import in May 2019, 5.45 % of papers were filtered out. The largest percentage filtered out was old historical data published before 1970, primarily caused by missing author information in the ADS records.

This software package, named The BAJer will hopefully prove itself useful in the future as a tool that provides more in-depth insight into astronomical journals' bibliometrics and a better understanding of possible metrics beyond the Journal Impact Factor. The value-added on top of the ADS is data cleaning and the reliable extraction of country data. Especially the country data can be quite untidy in the ADS database, and the search syntax is not adequate or too hard to deal with it for the average user. The easy-to-use interface is another value that The BAJer adds as it enables the average user to quickly gain insight into the droves of data stored in the ADS.

Application developers or data scientists who know SQL can create new plots beyond the one provided in The BAJer and extract other data with relative ease, given access to The BAJer's database. The API can be included in other automated workflows. The software has been released as open source[1].

1 https://gitlab.com/thebajer

CHAPTER 5

Outlook

For the future there are several areas where the project could be improved and expanded. One useful feature for the client would be the possibility to have multiple sessions that can be stored and quickly switched between.

Possible additions to the back-end are of course a greater number of end points delivering more insights with the according integration in the client to show it as a plot. Such plots are for example the display of late and early bloomers. This means how quick or late a paper starts to accumulate the majority of its citations after it has been published.

Analyzing the relation of keywords and topics is another big and interesting part. For this the aggregation of keywords needs to be improved in order to get good results. The same applies to deeper analysis into the role of authors. Both data sources suffer from the fact that they are free text fields which leads to problems as the same name or keyword can be written in many ways. Although the situation with author names could become easier if systems like ORCID are used throughout the ADS.

Making more use of the statistics the ADS already provides is another field where more features and improvements could be added. Though the validity of the data on which those statistics are based needs to be evaluated especially when considering historical data that is decades old.

Bibliography

Abt, Helmut A. (Jan. 2009): 'Do Astronomical Journals Still Have Extensive Self-referencing?' *PASP*, vol. 121(875): p. 73 (cit. on p. 1).

– (Aug. 2010): 'Have We Reached a Maximum Astronomical Research Output?' *PASP*, vol. 122(894): p. 955 (cit. on p. 1).

– (Feb. 1998): 'Is the Astronomical Literature Still Expanding Exponentially?' *PASP*, vol. 110(744): pp. 210–213 (cit. on p. 1).

– (July 2018): 'What Fraction of Papers in Astronomy and Physics Are Not Cited in 40 Years?' *PASP*, vol. 130(989): p. 074506 (cit. on p. 1).

ADS Bibcode (2019). URL: http://adsabs.github.io/help/actions/bibcode (visited on 05/13/2019) (cit. on p. 10).

Antonoyiannakis, Manolis (2020): 'Impact factor volatility due to a single paper: A comprehensive analysis'. *Quantitative Science Studies*, vol. 1(2): pp. 639–663 (cit. on p. 1).

Chawla, Dalmeet Singh (Sept. 2, 2020): *What's wrong with the journal impact factor in 5 graphs*. URL: https://www.natureindex.com/news-blog/whats-wrong-with-the-jif-in-five-graphs (cit. on p. 1).

Garfield, Eugene (1955): 'Citation Indexes for Science'. *Science*, vol. 122(3159): pp. 108–111 (cit. on p. 1).

– (Jan. 2006): 'The History and Meaning of the Journal Impact Factor'. *JAMA*, vol. 295(1): p. 90 (cit. on p. 1).

Gould, B. A. (Nov. 9, 1886): 'Preamble to the seventh volume'., vol. 7: pp. 1–1 (cit. on pp. 16, 23).

Henneken, E. A., A. Accomazzi, M. J. Kurtz, C. S. Grant, D. Thompson, J. Luker, R. Chyla, A. Holachek, and S. S. Murray (Apr. 2015): 'Computing and Using Metrics in the ADS'. *Open Science at the Frontiers of Librarianship*. Ed. by Holl, A., S. Lesteven, D. Dietrich, and A. Gasperini. Vol. 492. Astronomical Society of the Pacific Conference Series: p. 80 (cit. on p. 4).

Larivière, Vincent, Véronique Kiermer, Catriona J. MacCallum, Marcia McNutt, Mark Patterson, Bernd Pulverer, Sowmya Swaminathan, Stuart Taylor, and Stephen Curry (July 2016): 'A simple proposal for the publication of journal citation distributions'. Vol. (cit. on p. 1).

Levenshtein, V. I. (Feb. 1966): 'Binary Codes Capable of Correcting Deletions, Insertions and Reversals'. *Soviet Physics Doklady*, vol. 10: p. 707 (cit. on p. 9).

Michael J. Kurtz, Edwin Henneken (Oct. 22, 2018): *Citations to Astronomy Journals 2: Ranking the Journals*. ADS. URL: https://adsabs.github.io/blog/citations-journals-2 (visited on 05/10/2019) (cit. on p. 25).

OpenAPI Specification (v.2.0) (2019). URL: https://github.com/OAI/OpenAPI-Specification/blob/master/versions/2.0.md (visited on 05/18/2019) (cit. on p. 11).

'Time to remodel the journal impact factor' (July 2016). *Nature*, vol. 535(7613): p. 466 (cit. on p. 1).

List of Figures

List of Tables

Listings

A Appendix A

A.1 Table verifying extracted first author countries

The following papers have been randomly selected and the extracted countries manually verified against the affiliations stored in the ADS.

A partial match occurs if the affiliation contains more than one country. If no country could be extracted for the first author, the first hit on any subsequent author is used instead for the *first authors country* to have some country data. This is discussed in section 2.8.3 on page 21.

Table A.1: Extracted countries manually verified for randomly selected papers.
M. . . match
P. . . partial match
S. . . subsequent authors country used

Bibcode	Author	Country	M	P	S
2001MNRAS.327..799K	Kotov, O.	germany		1	
2008A&A...490..601S	Scelsi, L.	italy	1		
2008MNRAS.388.1487L	Li, Li-Xin	germany	1		
2011MNRAS.414.1667G	Guzman-Ramirez, L.	poland			1
2009MNRAS.395.2234Z	Zinchenko, I.	finland		1	
2019MNRAS.484.3921D	Davis, A.	canada		1	
2016MNRAS.460.1131S	Stonkutė, E.	lithuania		1	
2015MNRAS.453.2682I	Ineson, J.	united kingdom	1		
1996MNRAS.281.1352K	Knigge, Christian	united states	1		
2004MNRAS.351.1457S	Smith, Nathan	united states	1		
2018ApJ...863..101C	Chandra, Ramesh	india	1		
1994A&A...289..784C	Coe, M. J.	united kingdom	1		
2009AJ....138.1261F	Frayer, D. T.	united states	1		
2004ApJ...611.1200C	Chou, M.	taiwan	1		
2014ApJ...790...30J	Johnson, Marshall C.	united states	1		
2012ApJ...756..118B	Baraffe, I.	united kingdom	1		
1984ApJ...283..653F	Fix, J. D.	united states	1		
1986A&A...160..111G	Greggio, L.	italy	1		

Bibcode	Author	Country	M	P	S
2001A&A...366..157S	Skopal, A.	slovakia	1		
2016A&A...591A.138A	Ahnen, M. L.	switzerland	1		
1976A&A....53..329B	Baratta, G. B.	italy	1		
2015MNRAS.446.3725T	Toalá, Jesús A.	spain	1		
2009A&A...503..631A	Arlot, J. -E.	france	1		
2013ApJ...776...16P	Pucci, Stefano	italy	1		
2014ApJ...793..109J	Janowiecki, Steven	united states	1		
1989A&A...215..168B	Berton, R.	france	1		
1981A&A....98..344A	Angeletti, L.	italy	1		
1993ApJ...406..447T	Treves, A.	italy	1		
2012A&A...546A...2S	Sánchez, S. F.	spain	1		
1990A&A...230..220F	Fulle, M.	italy	1		
2014ApJ...780...58H	Henry, J. Patrick	united states	1		
1989A&A...209..244B	Becker, S. R.	germany	1		
2005AJ....129.1968L	Lockman, Felix J.	united states	1		
2018MNRAS.481..938Q	Qiao, Erlin	china	1		
2019A&A...623A.131K	Kislyakova, K. G.	austria	1		
2015MNRAS.446..470L	Luangtip, W.	united kingdom	1		
2002MNRAS.334L..27S	Shibata, Masaru	japan	1		
2017MNRAS.469S.755B	Blum, Jürgen	germany	1		
2018MNRAS.480.4154C	Cai, Xiaohao	united kingdom	1		
2001AJ....122.1238S	Shimasaku, Kazuhiro	japan	1		
2016MNRAS.459.3059S	Sadhukhan, Shubhadeep	india	1		
2018A&A...614A.115M	Magee, M. R.	united kingdom	1		
2016MNRAS.456.3004R	Rüdiger, G.	germany	1		
2012MNRAS.424.1244F	Fedeli, C.	united states	1		
1986ApJ...308..448L	Lindsey, C.	united states			1
1999ApJ...527..918W	Wyatt, M. C.	united states	1		
2010ApJ...720.1311H	Hiraki, Y.	japan	1		
2002A&A...383L...1C	Chang, H. -Y.	korea	1		
2001ApJ...546.1006B	Barrado y Navascués, David	germany	1		
1983ApJ...271..431V	Valdes, F.	united states	1		

Bibcode	Author	Country	M	P	S
2011MNRAS.413L..86C	Curran, S. J.	australia	1		
2010A&A...514A..15O	Onaka, T.	japan	1		
2002ApJ...581..550M	Muno, Michael P.	united states	1		
2019ApJ...871...55G	Grimaldo, Emanuele	austria	1		
2011AJ....141..144S	Stefanik, Robert P.	united states	1		
1991A&A...251..587R	Rossi, S. C. F.	brazil	1		
2013ApJ...778..178H	Hosokawa, Takashi	japan		1	
2013A&A...555A..65H	Hofmann, F.	germany	1		
2015MNRAS.451.1004E	Erroz-Ferrer, Santiago	spain	1		
1984A&A...130...53R	Rocchia, R.	france	1		
2013MNRAS.436..904B	Blackman, Eric G.	united states	1		
2016ApJ...824..136L	Li, Juan	china	1		
2008MNRAS.391.1191B	Bernardi, M.	united states	1		
2013A&A...554A..90V	Vranjes, J.	belgium		1	
2018A&A...614A.127L	López-Corredoira, M.	spain	1		
2000ApJ...528..979T	Templeton, Matthew R.	united states	1		
2018ApJ...868...23D	Drake, R. P.	united states	1		
2012ApJ...759...38R	Rapoport, Sharon	australia	1		
1997A&A...328..399W	Winter, O. C.	brazil	1		
2001ApJ...549..254S	Schinnerer, E.	united states	1		
2000ApJ...528..493T	Tirry, W. J.	united states	1		
2005MNRAS.357..929Z	Zappacosta, L.	italy	1		
2017MNRAS.472L..80L	Llinares, Claudio	united kingdom	1		
2014A&A...570A.131V	van Marle, A. J.	belgium	1		
2012MNRAS.421L..82A	Alentiev, D.	portugal		1	
2010A&A...510A...6B	Belkacem, K.	belgium	1		
2014MNRAS.438.3434R	Ruschel-Dutra, Daniel	brazil	1		
2016MNRAS.456.2611C	Cortesi, Arianna	brazil	1		
2017A&A...608A..99K	Kiss, L. L.	australia		1	
1986A&A...154..267L	Lucy, L. B.	germany		1	
1992A&A...264..379E	Escalera, E.	france	1		
1983A&A...126...59N	Nussbaumer, H.	switzerland	1		

Bibcode	Author	Country	M	P	S
2014AJ....147...55P	Pannuti, Thomas G.	united states	1		
1997ApJ...487..365A	Abt, Helmut A.	united states	1		
2011A&A...527A.144M	Maggio, A.	italy	1		
2007MNRAS.378....2S	Sim, S. A.	germany	1		
1988A&A...206..153G	Gail, H. -P.	germany	1		
2012MNRAS.424.1206W	Wyatt, M. C.	united kingdom			1
1998A&A...339..398I	Israel, F. P.	netherlands	1		
2007MNRAS.380..437P	Poole, Gregory B.	canada	1		
2016MNRAS.461.4206V	Verma, Kuldeep	india	1		
2007A&A...461L..13R	Recio-Blanco, A.	france	1		
2000MNRAS.315L..45L	Lahav, O.	israel	1		
2014MNRAS.443.3550C	Cruzalèbes, P.	france	1		
2013MNRAS.429.3288S	Sanderson, Alastair J. R.	united states			1
2013ApJ...767...87W	Wiener, Joshua	united states	1		
2015MNRAS.452.1841S	Shimizu, T. Taro	united states	1		
2008ApJ...678.1248V	Vrtilek, S. D.	united states	1		
1984ApJ...283..615B	Bell, R. A.	chile		1	
2006ApJ...651..688S	Shapley, Alice E.	united states	1		
		Sums	86	10	4

Acknowledgments

I thank João Alves for giving me the opportunity to write this master thesis at the Department of Astrophysics, University of Vienna, and the interesting hours of exploring bibliometric data and discussing the implications.

I would also like to thank all people who supported me in writing this thesis, especially my then girlfriend who in the meantime has become my lovely wife.

This research has made use of NASA's Astrophysics Data System Bibliographic Services.