**From Objects to Data Research Project – Final Portfolio 2 Oct. ‘14**

Arlette Mauritsz Berend Mul Koen van Schaik

**step 0: question**

Our original plan was to examine how different literary genres are distributed in the ‘book’ articles (more specifically, reviews) of the New York Times over time. It depended on the data we had acquired whether we could stick to this question, or if we had to modify our research question. As will become clear from this portfolio update, we have had no succes with our original plan, or our question for that matter.

We began looking at which structural data we could extract from the dataset articles and work from there. For now we were on a fishing expedition...

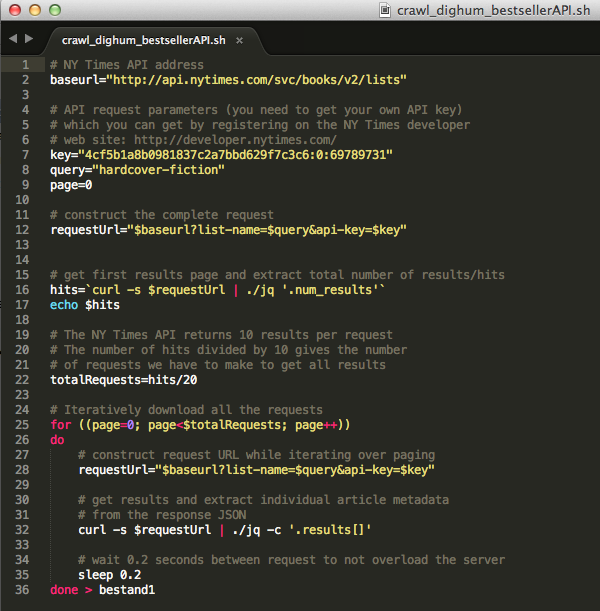
**step 1: acquiring data**

We searched in our ‘humanities’ database and saw that BOOK articles appeared as one of the highest categories (467)

**cat nytimes\_humanities\_tabbed.csv | awk '{print $3}' | sort -nr | uniq -c | sort**

But when exploring this data, we discovered that this wasn’t a really reliable data set for our research. When testing some URL’s of what were marked ‘Book’ and ‘BookReview’, -

**cat nytimes\_humanities\_tabbed.csv | awk '{if ($3 ~ BookReview) print}'**

-it turned that many of these articles were something entirely different. So we had to search for a new dataset, but we decided to stick with the New York Times. So we acquired new data from the bestseller API. We knew with certainty these extra articles had to be about books. We made a script to get the data, but we soon discovered that it only shows the bestseller list (10 to 20 books per category) of different, often overlapping categories.

So the information these data provided us was very limited. Crucially, the API doesn’t allow us to get all data of all years; it only contains data from the last 20 weeks.

We had to go back to the general Articles´ API, but we had to search for a very specific query that could give us a dataset of book articles. Queries like ‘book’ or ‘book reviews’ wouldn’t work, because it provides too much hits - over 600.000 in total. With the query ‘book review’ we also had way too many hits. We also encountered the same problems as in our humanities database: lots of the articles are not book reviews at all.

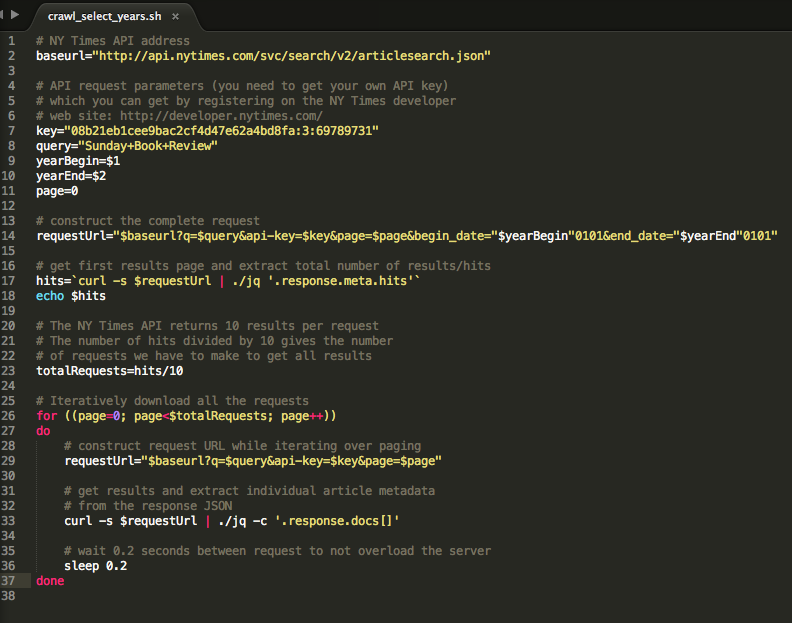
Accidently, we tested a URL and found a column, which maybe could’ve helped us out: ´Sunday Book Review´. These articles would certainly all be about books and the amount of hits seemed also more doable: 11.776. Testing some URL’s we saw that not all articles really came from the Sunday Book Review column and that there were some things we had to filter out of our database (e.g. a lot of ‘corrections’ to previous articles). But with the data that remained, it would be possible to perform interesting research.

Image 1) Crawl\_select\_years.sh

With our new query, we had to acquire the data from almost 12.000 articles. First, with help from our professor, we adjusted the crawl\_dighum script to our own query (see image, crawl\_select\_years.sh).

It would be too much to obtain all at once and it would be less risky to divide the data transfer in case of a lost Internet connection or other problems (all acquired data up to that point would be lost). With testing in the API Console, almost half of the articles turned out to be written after 2000. Therefore, we divided it in decades from 1850 to 2000 and from 2000 onwards we let the script run per year. Therefore, we made a script in which the crawl\_select\_years.sh was executed 29 times:

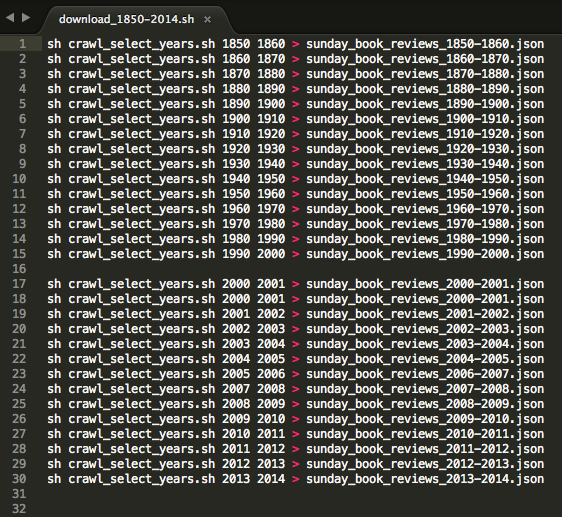


Image 2) download\_1850-2014.sh

This script was put in the command line

**sh download\_1850-2014.sh**

Because crawl\_select\_years.sh 2013-2014 defines as 20130101 – 20140101, we adapted this script as follows (see line 14, in image: crawl\_select\_years.sh):

**$yearBegin‘’&end\_date=yearEnd’’** (we removed the 0101).

Manually, we defined the date as 20140101 -20143112 to download all the data for 2014.

This process delivered us thirty different files. To collect these in one big file, we used the following command:

**cat Sunday\_book\_reviews\_\* > nyt\_Sunday\_book\_reviews\_1850-2014.json**

Everything seemed right with our new database, until we searched for the publication date with this command:

**cat Sunday\_book\_reviews\_1850-2014.json | ./jq .pubdate | sort –nr**

Every file, whether we choose a decade from 1850-2000 or a year from 2000-2014, showed only publications dates from 2012 onwards.

To verify if there was something wrong with our data set, we first checked whether the script we used to communicate with the New York Times API was working. We did this by comparing our script with the script the New York Times API Console gives (<http://developer.nytimes.com/io-docs>):

**q: Sunday book reviews**

**begin\_date: 19700101**

**end\_date: 198010101**

This delivered a result of 1211 hits and did contain publication dates of the requested years (it showed for instance 1969). We decided to mimic the request URL as accurately as possible in our script:

<http://api.nytimes.com/svc/search/v2/articlesearch.json?q=sunday+book+review&begin_date=19700101&end_date=19800101&api-key=sample-key>

To adapt our script to this URL we changed line 14 again:

**requestUrl="$baseurl?q=sunday+book+review&api-key=$key&begin\_date="$yearBegin"0101&end\_date="$yearEnd"0101"**

( 🡨🡪 Old line 14: )

**requestUrl="$baseurl?q=$query&api-key=$key&page=$page&begin\_date="$yearBegin"0101&end\_date="$yearEnd"0101" )**

This adapted script we saved as crawl\_select\_yearsVERSIE2.sh. Then we tested ‘crawl\_select\_yearsVERSIE2.sh 1970 1980’ and the old script ‘crawl\_select\_years.sh 1970 1980’.

Still the error of the publication date occurred.

Then we decided to compare the hits of the API Console on the GUI web console with the results of our API request script. Both gave 1211, so with high probability the results of the console and our script are the same. But the publication dates of the API remain incorrect (always 2012 >). We don’t know if we are doing something wrong or the data in the API contains an error (maybe it has something to do with holding on to the digital publication date?). At least the problem doesn’t occur in the NY Times Digital Humanities dataset…

**A OLD, NEW RESEARCH**

So our ‘acquiring data’ step seems to have failed. While we understand that these problems are naturally part of this kind of research, and that finding creative solutions is too, time was becoming very short (we encountered the pub\_date problem on 26Sept.). Because we were not able to gather useful data and we did not want to arrive with completely empty hands, we decided to go back to the \_humanities\_ dataset. Because we were more or less stuck on our original research question and we wanted to we provide at least some substantive and interesting results, we decided to, in the meantime, use a research question that would fit the \_humanities\_ dataset. Because Berend Mul’s research question fits the original data set provided by the course, we decided that this would be a good start to work on during the weekend:

With which academic field(s) are the humanities primarily associated (in the N.Y.T.), and how/has this changed through time?

From the research proposal we quickly extracted a list of tasks.

**Compiling a list of academic disciplines**

To compile a list of relevant academic field within the humanities, we decided to look at the Humanities/Liberal Arts departments of the top 10 American universities. We choose American institutions because our data set is from an American newspaper.

We decided to look which were the most influential universities, reasoning that these prestigious institutions would have more influence on perception. For this, we used the CWTS Leiden Ranking 2013. This world ranking looks at the influence of universities’ publications within the academic world. It includes a clear description of methodology and justification and it allows one to select scientific fields. Although this measures influence within the academic world and not society at large, the top listed university’s influence is still a better indication than a random guess from us would be. We choose the year 2013 instead of 2014 because this explicitly includes the humanities together with the social sciences (2014 does not explicitly list the humanities). In 2013 all top 10 universities within the field of the Social sciences & Humanities were located in the U.S. These were:

1. MIT (Massachusetts Institute of Technology)
2. Univ Chicago
3. Stanford Univ.
4. Harvard Univ.
5. Princeton Univ.
6. Yale Univ.
7. Darthmouth Coll.
8. Univ. Calif. – Berkely
9. Northwestern Univ.
10. Univ. Calif. – Los Angeles

Then we simply looked at the website for each university to determine how their humanities departments were categorized. Ideally we looked at the sub-departments, but sometimes the list of undergraduate majors provided better results (e.g. when departments carried an individual’s name).

Some universities combined their departments (‘Liberal Arts & Sciences'). In these cases we decided to omit obvious non-humanities disciplines but because wanted to be as unbiased as possible we did include ambiguous cases. In the table ‘Fields in the Humanities’ we made an overview of all studies, programs, majors etc. we found. From all these findings we deduced the more general fields in the humanities and checked whether each university included these or not (see table General Fields in the Humanities).

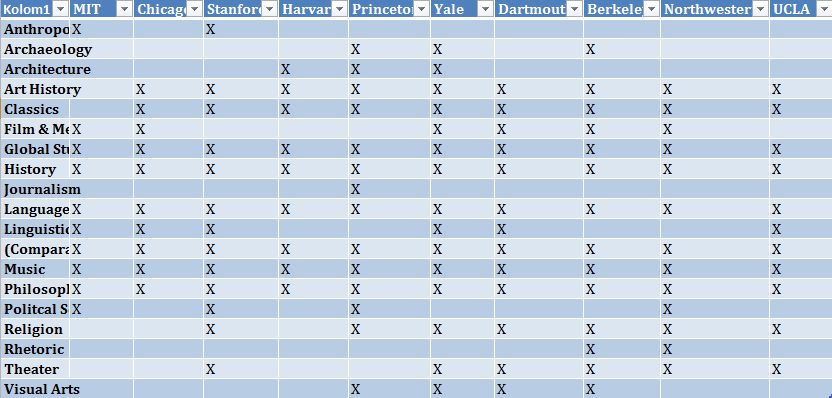


Image 3) Screenshot from the ‘General Fields in the Humanities’ table

Then, we had to choose which fields we were going to involve in our research. We choose to disregard some fields, because they are only a few times associated with the humanities; these included anthropology, journalism, political science and rhetoric. Other field could be put together in one category. Linguistics could be seen as a part of language studies, in fact it is the scientific study of language so in our search it will be a subcategory in the category language.

Some fields we have to disregard, because they are only few times are associated with the humanities: anthropology, journalism, political science and rhetoric. Thereby we propose to make ‘archaeology’ and ‘architecture’ subcategories of ‘art history’. These fields are quite interdisciplinary and have a lot of overlap. We had to take into account that archaeology, especially in America, is a field which is strongly associated with anthropology[[1]](#footnote-1). The NY Times confirmed this to us, by showing two kinds of archaeology articles. Those with the ‘’value’’ ART AND ARCHAEOLOGY and those with the ‘’value’’ ANTHROPOLGY AND ARCHAEOLOGY. For our research we will of course only use the first one.

The humanities fields we are going to search for are:

1. Art History

2. Classics

3. History

4. Film studies

5. Language

6. Literature

7. Music

8. Performing & Visual arts

9. Philosophy

10. Religion

But before we could simply search these categories, we had to think of subcategories. Some of the categories are obvious and are always used as such (e.g. history), but others need some extension in order to be complete (e.g. philosophy: ethics, metaphysics etc.) or just have synonyms. For categories like Language and Film we had to make important choices. When we should search independently for every language (English, French, Italian et cetera) or search just for ‘film’, this would deliver us huge amounts of results. These can’t be filtered out otherwise than checking manually whether each result is just a language/film, or if it really is the humanities field we’re looking for. Therefore we looked for other subcategories.

We came up with the following subcategories:

**1. Art History: history of art, archaeology, architecture**

**2. Classics: ancient Greek, Latin**

**3. History**

**4. Film studies: film studies, film, cinema**

**5. Language: linguistics, grammar**

**6. Literature: comparative literature, book studies, literary, poetry, prose**

**7. Music: musicology, acoustics, musical**

**8. Performing & Visual Arts**[[2]](#footnote-2)**: dramatics, painting, sculpting, musician, photography, theather/theatre, theatre studies**

**9. Philosophy: ethics, metaphysics, epistemology**

**10. Religion: religious studies, theology**

We will search for the category itself and for each of the subcategories.We also search for humanities general synonym, liberal arts. But making a script for this isn’t that easy as it looks. The commands are not that difficult, but we discovered that we had to work very precise to avoid duplicates.

**# grep\_AC-FIELD\_write\_to\_JSONfile.sh**

**FILE=$1**

**grep -i -w 'liberal arts' $FILE > arts.json**

**grep -i 'art histor\*' $FILE > art\_history1.json**

**grep -i -w 'art and archaeology' $FILE > art\_history2.json**

**grep -i -w 'architecture' $FILE > art\_history3.json**

**grep -i 'history of art\*' $FILE > art\_history4.json**

**grep -i -w 'classics' $FILE > classics1.json**

**grep -i -w 'Latin' $FILE > classics\_temp.json**

**grep -v -i 'America\*' classics\_temp.json > classics2.json**

**grep -i -w 'ancient Greek' $FILE > classics3.json**

**grep -i -w 'classical' $FILE > classics4.json**

**grep -i 'classicis\*' $FILE > classics5.json**

**rm classics\_temp.json**

**grep -i 'histor\*' $FILE > history\_temp.json**

**grep -v -i 'art histor\*' history\_temp.json > history1.json**

**grep -i -w 'antiquity' $FILE > history2.json**

**grep -i -w 'Middle Ages' $FILE > history3.json**

**grep -i -w 'Early Modern' $FILE> history4.json**

**rm history\_temp.json**

**grep -i -w 'film' $FILE > film\_studies1.json**

**grep -i -w 'cinema' $FILE > film\_studies2.json**

**rm history\_temp.json**

**grep -i 'language\*' $FILE > language1.json**

**grep -i 'linguistic\*' $FILE > language2.json**

**grep -i -w 'grammar' $FILE > language3.json**

**grep -i -w 'literature' $FILE > literature1.json**

**grep -i -w 'book studies' $FILE > literature2.json**

**grep -i -w 'literary' $FILE > literature3.json**

**grep -i -w 'poetry' $FILE > literature4.json**

**grep -i -w 'prose' $FILE > literature5.json**

**grep -i -w 'music' $FILE > music1.json**

**grep -i -w 'musicology' $FILE > music2.json**

**grep -i -w 'musical' $FILE > music3.json**

**grep -i -w 'acoustics' $FILE > music4.json**

**grep -i -w 'performing arts' $FILE > performing\_arts1.json**

**grep -i -w 'visual arts' $FILE > performing\_arts2.json**

**grep -i 'theat\*' $FILE > performing\_arts\_temp.json**

**grep -v -i 'Southeast\*' performing\_arts\_temp.json > performing\_arts3.json**

**grep -i -w 'photography' $FILE > performing\_arts4.json**

**grep -i -w 'musician' $FILE > performing\_arts5.json**

**grep -i -w 'dramatics' $FILE > performing\_arts6.json**

**grep -i -w 'sculpting' $FILE > performing\_arts7.json**

**grep -i -w 'painting' $FILE > performing\_arts8.json**

**rm performing\_arts\_temp.json**

**grep -i 'philosoph\*' $FILE > philosophy1.json**

**grep -i -w 'ethics' $FILE > philosophy2.json**

**grep -i -w 'epistemology' $FILE > philosophy3.json**

**grep -i -w 'metaphysics' $FILE > philosophy4.json**

**grep -i -w 'religion' $FILE > religion1.json**

**grep -i 'religious\*' $FILE > religion2.json**

**grep -i 'theolog\*' $FILE > religion3.json[[3]](#footnote-3)**

The precise thinking already started with the first category ‘art history’: by making the search ‘art histor\*’, the command would also include art historians. These are all terms we needed. The second difficulty came with subcategory ‘Latin’. Just searching for Latin also included all Latin American results. Therefore we decided to, immediately after putting these results in a temporary map, include a grep –v ‘America\*’ command to get rid of these. We searched for ‘ancient Greek’ to prevent the Modern Greek language/churches/communities et cetera to come in our file.

With searching for ‘histor\*’ we had to realize that these would again include the ‘art histories/historians’. That’s why we are forced to run a grep –v ‘art histor\*’ command here. This way we keep the two categories separated.

Under the category ‘film studies’, searching for the subcategory ‘film’ would be enough to collect data, because these result will also include the main category ‘film studies’. The same is true for the ‘theater/theatre studies’: searching for theat\* is enough. But with searching for theat\* we encountered something weird: it selects also ‘southeast’. We had no idea why, but to remove these we again run a grep –v command.

As can be seen, on other places we also added the \* to the search to cover also the plurals and adjectives.

With this script it must be able to collect all data needed. The different files that come out we combine to one file per category with cat ...\*.json > ...\_complete.json.

Although our scripts may not be the most difficult ones, we learned that acquiring data demands very precise and logical thinking. We kept changing our script, because every time someone notices another detail that will yield duplicates. With our final script we hope to have covered most of the difficulties.

**Step 2: Parsing**

Another step was needed before we could continue our research. Because the searches for subcategories may deliver duplicates in the combined file, we executed a ‘sort –u’ command for each category:

# sort -u complete\_file sorted\_file\_JSON.sh

sort -u art\_history\_complete.json > art\_history\_sorted.json

sort -u classics\_complete.json > classics\_sorted.json

sort -u film\_studies\_complete.json > film\_studies\_sorted.json

sort -u history\_complete.json > history\_sorted.json

sort -u language\_complete.json > language\_sorted.json

sort -u literature\_complete.json > literature\_sorted.json

sort -u performing\_arts\_complete.json > performing\_arts\_sorted.json

sort -u philosophy\_complete.json > philosophy\_sorted.json

sort -u religion\_complete.json > religion\_sorted.json

This command indeed removed the duplicates in our files, but we noticed, by looking in the CSV files in Excel, that some duplicates still remained. These are mainly ‘Death Announcements’ and other announcements, which were probably not removed by the command because the first one is published at midnight, the second a few hours later. So, because we can’t remove these with commands and it is really difficult to parse these manually, we have to keep in mind that there may be some duplicates in each category.

For the category ‘classics’ we had to perform an extra command. Because ‘classics’/‘classical’ are broad terms, the grep command also delivered a lot of ‘classics’ in music. Therefore, we decided to run a grep –v –i ‘music\*’ command on the ‘classics\_sorted.json’ file to get all the music articles out of it. With this we killed two birds with one stone: the Latin (American) music was also removed. Still there will be some ‘wrong classics’ in our file, as there will some flaws be in every category, but a quick look through the results showed that the majority consisted of the right classics and Latin terms.

**Step 3: Filtering**

With the files sorted and parsed, we could start with filtering the files to get the needed information out of them. We decided we wanted to know two things for each category: what is the distribution of their articles over time?; how are the main category and the subcategories distributed in the file?

For the distribution over time we wanted to make graphs, so we saved our results in CSV-files. We run for each category the following script (art history is taken as example here, see italics):

# from\_sorted\_to\_fq\_yrs.sh

cat *art\_history\_*sorted.json /jq .pub\_date | sort > *art\_history*pubdates.csv

cat *art\_history*pubdates.csv | awk -F '\-' '{print $1}' | uniq -c > *arthistory*frequences\_years.csv

cat *art\_history*frequences\_years.csv | awk '{print $1}' > *arthistory\_*frequencies.csv

cat *art\_history*frequences\_years.csv | awk -F '\"' '{print $2}' > *art\_history*years.csv

Firstly, we asked for the publication dates of the articles with the ‘jq’ command. These we sorted chronologically and saved these in a separate CSV file. Then we wanted to keep the years only and count the frequency of the articles per year: we had to print the first column by awk and sort and count these with ‘uniq –c’. These results we again saved in a separate file. Thereafter, we needed to divide this file in two separate files, one with the frequencies and one with the years (this way we could make our graphs the easiest). So, we saved the first columns of the ‘...frequences\_years.csv’ and the second column in two different files. These two csv files we combined with making graphs in Word. We copied and pasted the years in the first column and the frequencies in the second column: this way we put the years on the x-as and the frequencies on the y-as. The result can be seen in image 4.

For the distribution of category and associated subcategories, we decided to make bar graphs. These were quite simple to make: we just counted with the ‘grep’ command the occurrence of each term and put the results in the graphs (see image 5). Not with art history, but in other categories, there were overlapping terms. Therefore, we had to take into account that we should apply some (simply) calculations. For instance, searching for main category ‘literature’ also includes subcategory ‘comparative literature’. So the result of comparative literature had to be subtracted from literature to remain the right number.

Image 4) Time graph of Art History: distribution during the years Image 5) Bar graph of Art History: distribution of main category and subcategories

**Doing a search in the dataset for each term & counting**

First we decided to check how many times the most important terms were present in the database. For this we wrote a separate script and called it ‘**count\_presence\_academic\_field\_define\_file.sh’**. First it lists the terms searched for and then it counts each term with ‘grep --count’ in the file that can be separately defined by the user in advance. Naturally we want to list the right academic field besides the right count results in a column, but we have not yet found a way to do this. In the command line we typed:

**sh count\_presence\_academic\_fields\_define\_file.sh nytimes\_humanities\_tabbed.csv > fields,count.txt**

:

Image 3) count\_presence\_academic\_field\_define\_file.sh Image 4) fields.count.txt

\* By the Word spelling checker, we discovered that the

British spelling ‘theatre’ also delivers 22 hits. These will

also be surveyed in our research.

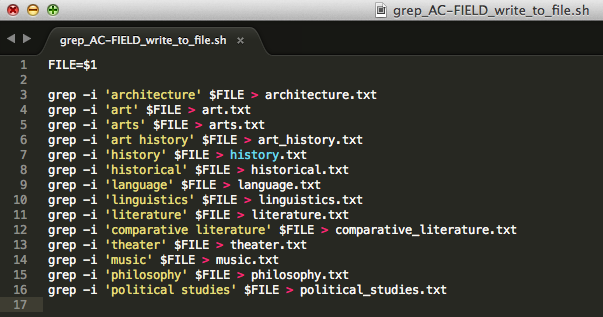
As a second step we decided to make separate files for the results that contained these terms. We excluded ‘theater studies’ for obvious reasons.

Image 5) grep\_AC-FIELD\_write\_to\_file.sh

It simply searches for articles that contain the term (and with ‘-i’ disregards cases in the data set) specified by the user and writes these to separate .txt files bearing the name of the respective term.

With these files, we can do more interesting research. For example, we wrote a script that prints the publication date and print page for all articles containing a specified query and data set:

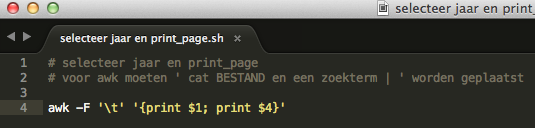
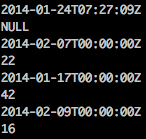


Image 6) selecteer jaar en print\_page.sh

E.g.

**cat nytimes\_humanities\_tabbed.csv architecture | sh selecteer\ jaar\ en\ print\_page.sh**

provides (small sample):

We also made small Dates\_keywords.sh script that could isolate the publication date and keywords from the \_flat.json file.

**cat nytimes\_humanities\_flat.json | ./jq [".keywords"],[".pub\_date"]**

It is limited to the \_flat.json file, for this data set contains keywords.

If we continue with this research question and dataset, we still have many tasks to do, for example:

* Further scrutinize and focus our search terms
* Gather more details about articles:
  + keywords
  + article length
  + separate per section and decide how each section is rated
* Find out whether we can separate the search terms as academic fields from their general meaning (count a sample by hand and calculate this for the whole set), and decide whether that is even desirable.
* Incorporate all this information and come up with a clear way to visualize it.

For now it is interesting to see that there is a large difference between the presence of the different academic fields within the data set. Two terms especially stand out: Art and Arts. Art is obviously a word that can be used in many different contexts so we have to find a way to isolate its meaning as an academic field. Counting a 100 articles by hand and then calculating the ratio could be one possible solution. Arts is more or less a synonym for the humanities, and in hindsight it probably could have been excluded from the list. History also scores rather high with 882 hits. Just as art, history also has a very general meaning, but evidently the past is of rather minor importance compared to art in this data set. It is also interesting to say that the political sciences are barely mentioned. This field of study traditionally belongs to the social sciences, but sometimes it is mentioned in relation to the humanities too (with which it has a large overlap). As it turns out, the New York Times does not really mention it often in relation to the humanities. Lastly, most universities listed ‘comparative literature’ as their field of study (in contrast to ‘literature’). In the data set this is the complete opposite: literature has 408 hits, comparative literature only 27. Below are some visualisations of the data.

Obviously these are just preliminary searches, but it is interesting to that, at first glance, there are such large differences.

Grapgh 1) Total no. of hits

Graph 2) excluding ‘Art’, ‘Arts’

1. C. Renfrew & P. Bahn, *Archaeology: Theories, Methods and Practice,* 2012, p. 12. [↑](#footnote-ref-1)
2. Courses in here could be compared with the Dutch Art Academies; it is the more practical side of Art, which turned out to be closely associated with the Humanities studies at the American Universities. [↑](#footnote-ref-2)
3. We created the same script for the CSV file. [↑](#footnote-ref-3)