University of Duisburg-Essen
Faculty of Business Administration and Economics
Chair of Econometrics

Whats is driving Chinese News?

Linking Daily Text Data with Economic Data

Scientifc Report

Submitted to the Faculty of
Business Administration and Economics
at the
University of Duisburg-Essen

from:

David Schulze, Eyayaw Teka Beze, Nils Paffen

Matriculation Number:	3071594, ID David, ID Eyayaw
Study Path:	Phd Programm RWI, VWL M.Sc.,
Reviewer:	
Secondary Reviewer:	
Semester:	3rd Semester
Graduation (est.):	Winter Term 2020
Deadline:	Apr. 7th 2020

Contents

List of Tables

List of Figures

1 Introduction

1.1 Motivation

Ever since it's establishment in 1948 as the Chinese Communist Party's (CPC) officially designated publication organ, and especially since it announced the foundation of the People's Republic by Mao Zedong on October 1st, 1949, the People's Daily newspaper (also known as Renmin Ribao or RMRB) has been an object of the highest interest for anyone interested in modern China. It is safe to say that it's front page is that of the most widely published newspapers in Chinese and it's the most widely read. This amount of attention and it's clear designation as voice of the CPC have made it an invaluable source for information on China's ruling elite's communication with the masses. In times of crisis, even tiny changes in placing, formatting or wording are chosen and interpreted with extreme scrutiny **tan1990**. The reason for this is on the one hand the need for some form of discussion involving the government and the educated citizenry **kuhn2002**, but also the sensitivity of certain topics, also called censorship.

In the past, close reading and an intimate knowledge of the Chinese language were the only tools available to researchers interested in this publication. Even the most well-read scholars of modern China will have to admit that reading the paper daily and in it's entirety, even just the front page, will be a thankless undertaking. Most articles are official statements and collections of facts about the activites of leaders, or plain positive messages about some aspect about the nation, also called, by the CPC themselves, propaganda. The nuances that to detect require years of study are difficult to check against factual evidence, short of spending hours of reading yourself. Only rarely are messages communicated as clearly, as back on the founding day of the People's Republic.

This presents in our view a very urgent opportunity for automated data mining. Unlike historic literary corpora like the works of Shakespeare that can be analysed by generations of scholars, news is by its nature fleeting and often needs to be analysed in a hurry and theories tested as events unfold. Fortunately, the People's Daily offers the whole paper's content every day for free on its homepage, paper.people.com.

To test our idea for an automated evaluation of People's Daily articles on their own and in context of economic data, we proposed the development of an app, that would automate the task of updating the news corpora and economic data, as well as some descriptive and basic analytical steps used in quantitative text analysis. To make these results available to non-Chinese speakers, we include a simple translation routine, that gives the most common word for word translation, even if not the meaning of entire

articles or sentences.

We stress that this will in no way substitute any qualitative reading, knowledge of Chinese politics, expertise in Chinese language or even text mining of the news in general. But to be able to quickly develop and test quantitative hypotheses about Chinese news and its relation to economic data, might be a useful tool on the way to further research and in-depth text mining.

1.2 About the People's Daily newspaper

For the app we focus on the first two pages, because the first is arguably one of the most important daily publications in China, and the second might offer interesting contrasts, because it is aimed at a different audience, while significance diminishes rapidly with increasing page count: The latter pages of this leading CPC publications are actually full-page advertisements for private company's products, among others.

The first page's layout is different every day, according to the needs of the editors: The documentory function implies that a lot of information is squeezed into a tight space. Articles may be reduced, truncated or pushed to the side of the page to make room for symbolic pictures [1] and voluminous leading articles. On the other end of the spectrum are crowded front pages with up to 15 tiny articles commemorating each meeting of a tightly packed international summit schedule, with 15 headlines reading: "Xi Jinping meets (insert name of foreign national head of state)" [2].

How can we ever expect to extract useful information from such a data source? Well, for nuances in reporting to have any impact, they have to be special or deviating in some way from an established norm. By gathering data over hundreds of days, we hope that these patterns and deviations will become visible. Also, comparing subsets of data such as front and second page, reporting before and after a certain date, commonalities will be filtered out and differences highlighted. Since our data cover the year 2019 and the first months of the Covid-19 outbreak that was first documented officially by authorities in Wuhan, Hubei Province China on 2020-01-05 [3], this constitutes an opportunity to evaluate how this major shock is covered and news compare before and after the outbreak was acknowledged.

While text mining and natural language processing algorithms have matured extremely in the contexts of machine learning and applications for example in online search, the

[1] See for example the pictures displaying national remembrance of the "martyrs sacrificed in the struggle against Covid-19" with bowing national leaders and the lowered flag in black and white front page from 2020-04-05

[2] front page from 2020-04-26

[3] "Wuhan Health Commission report on the situation concerning a viral lung disease with unknown origins", wjw.wuhan.gov.cn

technology barrier has limited it's adoption in the social sciences. To make basic results accessible and reproducible, is finally a large motivation for the creation of this app.

1.3 Overview

In the next sections we will introduce the general concept behind the construction of our app, its functionality, and our general process in coding the individual steps.

2 Construction of the Application

2.1 General Concept

The functionality of the app has three major parts:

- updating databases with text and economic data
- handling translation and transformation of data
- outputting descriptive and analytical information

2.1.1 Shiny user interface

This was implement using the R package *shiny*, especially *shinydashboard*, to create a layout with a tab menu on the left hand, and a space for content on the right. Large buttons convey key information quickly, and dynamic boxes can be filled with text, tables, buttons, drop-down menus, and others. Especially the feature to have tabs on top of the box to show for example several plots in one space, was very useful.

2.1.2 The method

Formally, the functionality puts our app in the toolbox of so called *distant reading* approaches in methods of text analysis, the use of which in social sciences in general and economics in particular is growing.. As discussed in this workshop paper by **scrivner_davis2017´**, this method is not designed to substitute or render *close reading* obsolete. It actually increases the value of such traditional methods involving a detailed analysis of single texts, by pointing out interesting texts and features in a growing sea of available texts. To think of an overly simple example, compare reading a list of references to reading every single article contained therein. The list does not give the same information, because it abstracts from the detailed texts. But with the

list, we can find the texts we are looking for, saving time and leading to new insights, i.e. texts we might have missed.

The **updating** gives the user the ability to gain instant insights in developments as they are unfolding, as soon as data are available. The People's Daily website is updated regularly and many other other series reachable via API are reliable sources for economic data new enough to make this a useful feature. It is possible to extend the app to include monthly and yearly economic data as well, but this would be more useful, if the text data would be extended longer into the past to match.

More precisely, our app contacts the servers of the Peoples Daily website. and economic data APIs to compare the information with our database and check if new articles are available. If the answer is yes, it downloads them for research purposes without intent on distribution or commercial use and expands our database accordingly. You can find the raw .rds-files created by scraping these articles in the folder "/data". Our first attempt created separate files for years and pages, to decrease size and impact of errors. Ultimately, with additional time, creating for example a unified SQLite database for all objects would be computationally more efficient and lead to a simplification of all the following code.

The **translation and transformation** aspect makes the data more useful by offering a translation and transformed version suitable for plotting and analysis. Having a word-for-word translation of the newspaper data allows users to gain a limited impression of the content, allow for quick scanning of the text by users not sufficiently proficient in Chinese, and also allows for streamlined mining. In particular, the free Yandex translation API used here, translates several versions of a Chinese word to the same English correspondence. So this n-m matching, where n>m, implies that a user who searches using the English word will get more correct hits than for one particular Chinese word, without necessarily losing the meaning. One of our team members is proficient in Chinese and spot checks have confirmed the quality of the data to be sufficient (most mistranslations concern less common names and obscure phrases). For literary allusion and poetic nuance, of course, the translation will necessary and always fail.

Of course all the limits of analysing language quantiatively also apply in Chinese: Rarely is understanding a text as easy as grasping the meaning of the individual words. For example, when searching for the literal translation of coronavirus, recent articles will yield surprisingly low hits. The reason is that like a lot of Chinese texts, the news uses shorthands or less direct terms, especially for a subject that is politically and emotionally sensitive like this one. Therefore a better search would look for the Chinese equivalent of "outbreak", which in the first months of 2020 is only understood in one way, as a reference to the pandemic.

This does not affect general inquiries, about the most frequent words for example, in the same way. Censorship and propaganda are more obvious on the aggregate level. It shouldn't surprise anyone to see that the name of president Xi Jinping, the CPC and other government organs feature highly among the most common words. This is not a mistake, but rather revealing an important aspect of the People's Daily's reporting!

All unique Chinese article words, excluding symbols and numbers, were translated and saved a bilingual dictionary in the folder "/output" as a .rds-file and in comma-separated values, to allow use in other contexts. We have gathered around 95000 unique terms over around 15 months. For comparison, the free online dictionary MDBG.net is based on the CC-CEDICT database which as of 2020-04-06 counts around 120000 entries, so this does not seem entirely unreasonable. In the same folder, processed and translated copies of the articles are saved. The process uses an NLP algorithm

Outputting finally is the essential part of the app: If users can't see the result of updating and transforming the data, the app won't be of much use. Ultimately, it would be most desireable to offer a high degree of customization in using the app for output: Choosing search words, specifying paramters, etc. would greatly increase its usefulness. In the end, gathering and transforming the data proved most time-consuming. This was to be expected, but it meant that we had to discard our most ambitious ideas for describing, analysing and visualising the data as well as making it interactable.

In summary, as a proof of concept, this app should show a potentially useful application of using R with the shiny web-app, the ggplot and other tidy packages, API connections, scraping and data mining in an applied context.

2.2 Webscraping the People's Daily archive with Rselenium and Docker

The platform CrossAsia is a service provided by the Staatsbibliothek zu Berlin and funded by the DFG. It collects and offers access to data for many kinds of Asia-related research. Through CrossAsia we have access to a database of issues of the People's Daily starting from 1946. The oldest edition is dated on May 15, 1946. The People's Daily website on the other hand offers issues since the beginning of 2019 and is offered directly to everyone on the internet.

To combine the long archive and the easy access to the new issues, we had the idea to develop a webscraper that updates using the People's Daily website and adds the articles to our database based on the archive via CrossAsia.org.

Since this archive is only accessable with an account and a memebership, for the

scraping of the archive, a login for the scraping process is necessary. The best solution to send the login credentials, browse through the archive, and scrape the content, we came up with is an Firefox image in Docker accessed by the RSelenium package.

Docker is a batch of so called platforms as a service (PaaS), which is a type of cloud computing service to manage and run application without any kind of infrastructure. For our purpose we used Docker to build a browser in a virtual machine and then connect these with R using the RSelenium package. For a detailed installation guideline we recommend to follow this. To interact with the virtual browser image in R, RSelenium uses a class called RSelenium::remoteDriver, which features several functions such as *remDr$navigate("url")* to browse to a specific website or more specific ones like *remDr$screenshot()* to display or save and screenshot of the actual webpage, or *remDr$sendKeysToElement()* which allows the user to send specific keys like "Enter" to the browser.

Before we can start to scrape the articles we need to start the Docker image as explained in the guideline. Then the first thin the scraper function *ca_scaper()* function calls is *ca_login()*. The latter connects the remoteDriver with Docker. When this is done it navigates to the login page of CrossAsia.org and hands over our account credentials.

In this case, 19460515 for the date and 1 for page 1. The link structure of the archive differs significantly from that of the free archive. Besides a personal session token, in this case "s894ib9c043f", the link structure differs by the access points to the archive. In addition to CrossAsia.org, the Staatsbibliothek zu Berlin should be mentioned here in particular. This can be found in the suffix ".erf.sbb.spk-berlin.de". The article overview of each issue is uniformly accessible via the date and page number.

In this case, 19460515 for the date and 1 for page 1. 32-character ID is used for article archiving. The latter shows the oldest article of the archive. Since the ID does not seem to follow a fixed pattern, it must be obtained from the article overview for the scraping process. A list of all article IDs can be retrieved on each overview page by using the HTML node "li h3 a" and the attribute "href". Now links can be generated from the date format and the IDs to call up the individual articles. To fullfill this task the scraper calls the function ca_date_urls. The so created url list is then passed to *ca_content()*.

Afterwards the scraper maps over this list while calling the *ca_content()*-function. Almost all important information such as author, title, subtitle and article content can be retrieved via the html node "#detail_pop_content" and the underlying nodes ".title", ".subtitle", ".author", and "p". After scraping the information all content is saved in a tibble.

2.2.1 Page protection

Since CrossAsia.org is paying the provider of the archive a fee for the content they give acces to, both they and the provider are oriented towards protecting the archive data from illegal downloads. Although the use for preview purposes is allowed, the user cannot always be distinguished from a "data pirate", i.e. offering the articles on the Internet after downloading them. In addition, the provider wants to keep his server load as low as possible in order to grant as many users as possible quick access to the archive at the same time. If several users now send automated requests to the archive, this can affect the speed of the servers. By randomly generating sleep times in the code, the scraping behavior can be adapted to the usage behavior of a human being. CrossAsia.org asks the user to solve a simple captcha after a certain number of article calls. To solve these problems, our webscraper first checks for a location in the HTML code that is present on both the article page and the captcha page. If the site structure equals the one of a saved version of the captcha site the function *ca_captcha()* is called.

First, a screenshot of the page is saved by the remoteDriver and cropped in a way that only the captcha image is left. The page 2captcha offers the possibility to solve captchas automatically for a small fee of less than 1$ per 1000 captchas. For this purpose, the user is provided with an API key, which can be used to transmit the image information as a POST request with multipart-encoding. Afterwards, the user is assigned a ticket-ID, with which he can retrieve the solved captcha code as a string after about 5-10 seconds of waiting time. Since the operator may sometimes experience increased solving times, we check the content of the string. A waiting time of 5 seconds is added if the captcha has not been solved yet and an attempt to grab the code follows the additional waiting time. The captcha-code is now sent to the to the corresponding form of CrossAsia.com and will be forwarded automatically. After entering the captcha code, we check again if the forwarding was successful or if the entered captcha code was invalid and therefore another captcha must be solved. Even solving these captures codes and extensive waiting time of about 10 minutes after each page, we expierenced that the archive operator gives a time-out error to users additionally, if they try to look at too many articles in a given time. The time-out length and the length until a timeout is called differs from 30 minutes to a few hours. We tried to check for this site behavior but were not completly succesfull in providing a function to check for the time-out and wait until it disappears.

2.2.2 Output

In the end the intense timeout periods lead us to the decision that the webscraping of the archive is possible but extremely time consuming. The code will be able to scrape

the content as described, but for the purpose of the app we decided to not integrate the archive content so far.

2.3 Translation from Chinese

To make the app basically usable for non-Chinese speakers, a translation feature was introduced. Because Chinese has no spaces, we needed to use a NLP algorithm, so that in the article texts, spaces could be placed between words to separate them. Unique words were added to a Chinese dictionary. Each word was in turn passed on to the translation API from Yandex. This yields in most cases a reliable one-word translation. Alternative translations were beyond the scope of this app, and for in depth analysis, a knowledge of Chinese or at least use of more translation tools is necessary. This dictionary is exported in .rds and .csv files for further use. The articles are then translated word for word. This will speed up using the app for text mining with English words, even if it does not give a useful translation of the complete texts.

2.3.1 Some additional remarks:

To reduce use of the Yandex translation API (which has limited free bandwith) in each update, only new words are translated and added. This allowed an approximation of the range of unique words used. For illustration: Translating all the unique words (words includes here names, idioms, technological terms, etc.) yielded at least 50150 unique words for the first pages of all of 2019. The second pages of 2019 added 16393 new words. For the first three months, we found still 3228 new words for page 1 and 2350 new words for page 2. Even when allowing for misclassification by the NLP algorithm (sometimes words might be recorded as separate and again as a compound), this suggests that the news needs plenty of new words to adapt to current developments, e.g. “Covid-19”.

Another aspect of the dictionary: While all Chinese words are unique, the English translation yielded around 25000 duplicated entries. This means that users searching for example for “satellite” will find hits for “Wei Xing”, “Ke Weixing” and “Renzao Wei Xing”, which all carry that meaning, improving the accuracy of text mining.

2.4 Word frequency time series and economic data

In order to gain an impression of the impact the frequency of a single word can have on the economy, we created the function *ts_word_frequency()* to generate time series and visualize the results. The user first specifies whether he wants to search the articles on

page 1 or page 2 (page_num) for a specific word (eng_word). In addition, the user can specify a time period in which the analysis is to be performed using start_date and end_date. In the parameter econ_data he also determines whether he wants to compare the word frequency of eng_word with the NASDAQ_CNY or dollar_yuan_exch, which have already been explained in the section on economic data. We have integrated a function to download them from different sources and then adjust and, if necessary, normalize them according to the user's input time parameters. Furthermore there is the possibility to extend the frequency of time series from monthly to daily. It is also possible to extend the frequency of time series from monthly to daily.

2.5 Economic data retrieval

To give the user an impression how a term can impact the economy we decided to implement two indicators for chinese economy. The NASDAQ CNY is a daily stock market index of 74 international chinese companys. The other on is the exchange rate of Chinese Yuan to one US Dollar.

2.6 Exploratory statistics, visualisation, and text mining

3 Description of Data

The data are collected from the Chinese People's Daily newspaper for year 2019 and 2020. The daily newspapers are published on a consistent structure (we give the details of the structure below) online on the newspaper's website. Despite the slow loading speed of the website, We tapped into its structure and have scraped the **first two pages** of the daily newspapers.

For instance, the 1st page of the issue published on *2020-03-22* can be accessed here.

Figure 1: Example newspaper page published on 2020-03-22 (left), and Section or column 1 of it enlarged (right).

We dissect the different characteristics of this particular article as follows and these properties apply to all other articles.

First of all, the main link to the article is this one: (http://paper.people.com.cn/rmrb/html/2020-03/22/nbs.D110000renmrb_01.htm). This takes us to one page of the newspaper, i.e., to the 1st page of the newspaper in this particular example. However, a lot of content is packed on this single page. There are 10 different sections or columns on this article (see Figure 1 for this example newspaper page published on 2020-03-22).

- The prefix of the URL, i.e., **(http://paper.people.com.cn/rmrb/html)** is the same for every article, regardless of page number or edition.
- Date on which it was published: **2020-03/22**
- Page number of the newspaper: **01**
- Section id prefix: **nbs.D110000renmrb**.

As we can see in Figure (1) there are several different sections crammed or squeezed into the single page of the newspaper and these parts (sections) are clickable, and each has a unique id. A click on each section will redirect to a link where one can access the full content of the section in an enlarged view. For example, the first section out of the 10 sections on this example article is section 1 or (see Figure 1 –right). The section id for this first section is **nw.D110000renmrb_20200322_1-01**.

Accordingly, the id of each section on an article is of this form: **nw.D110000renmrb_yyyymmdd_page#**. Each section of the single page newspaper has the following additional characteristics.

- Title (**h1 tag**)
- Subtitle (**h3 tag**)
- Number of paragraphs on each page, and
- Content or body of the news article section.

The maximum number of sections on a page is 9, in an article published on 2019-04-26, and the minimum is 1. On average, there were 6 sections per a page of an article, for the newspapers published since January 1st, 2019 regardless of the page number. Looking at the frequencies of articles individually, the tendency is for page 2 to have less issues with very high numbers of articles above around 9.

Therefore, the contents of all the sections (**paragraph or p-tags**) together–on the page of the news article– make up the contents of the entire (single) page–which are

compactly placed in a single page of the article. Particularly, we scraped the sections of the newspapers–through their unique ids. The date and the page number of the newspaper uniquely identify a newspaper, where the combination of which forms the ids of the sections–prefixed with **nbs.D110000renmrb**.

Moreover, most of the news articles are bulky in terms of number of paragraphs and text volume. There are 70.23 number of paragraphs per page, and 15.87

4 Descriptive Statistics

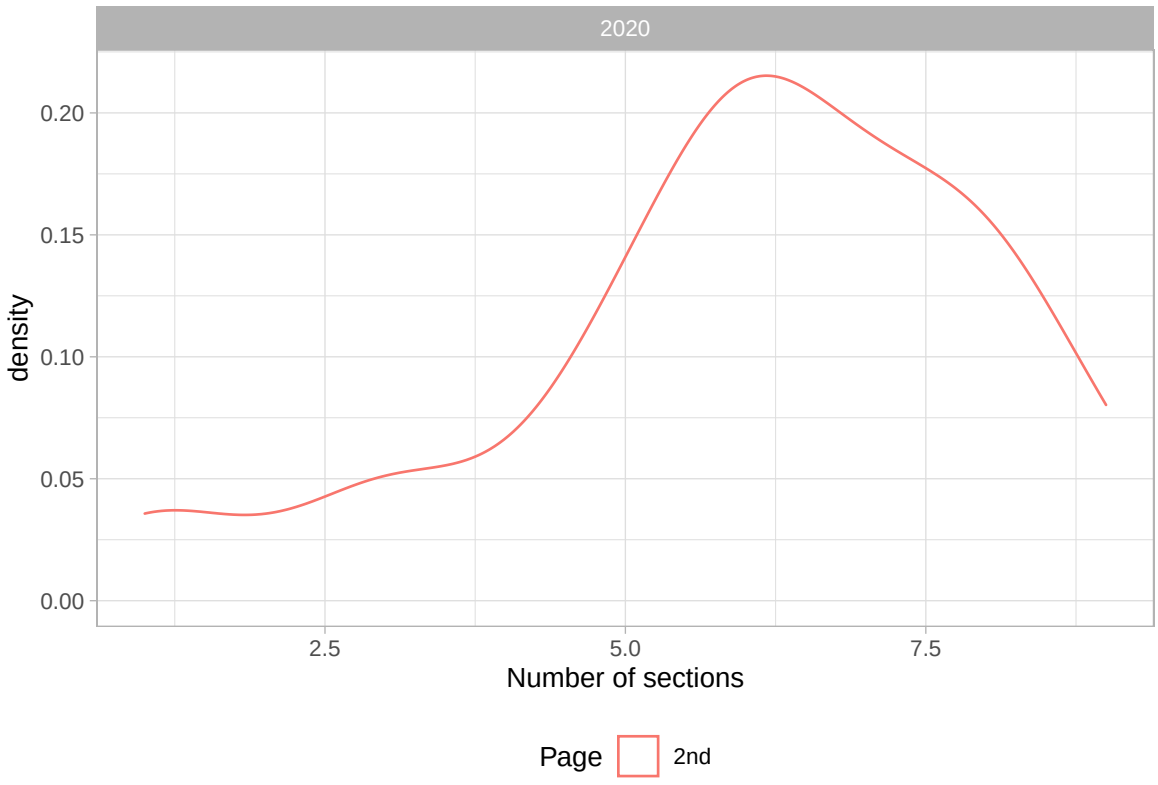


Figure 2: Distribution of **number of sections** in a page of a newspaper

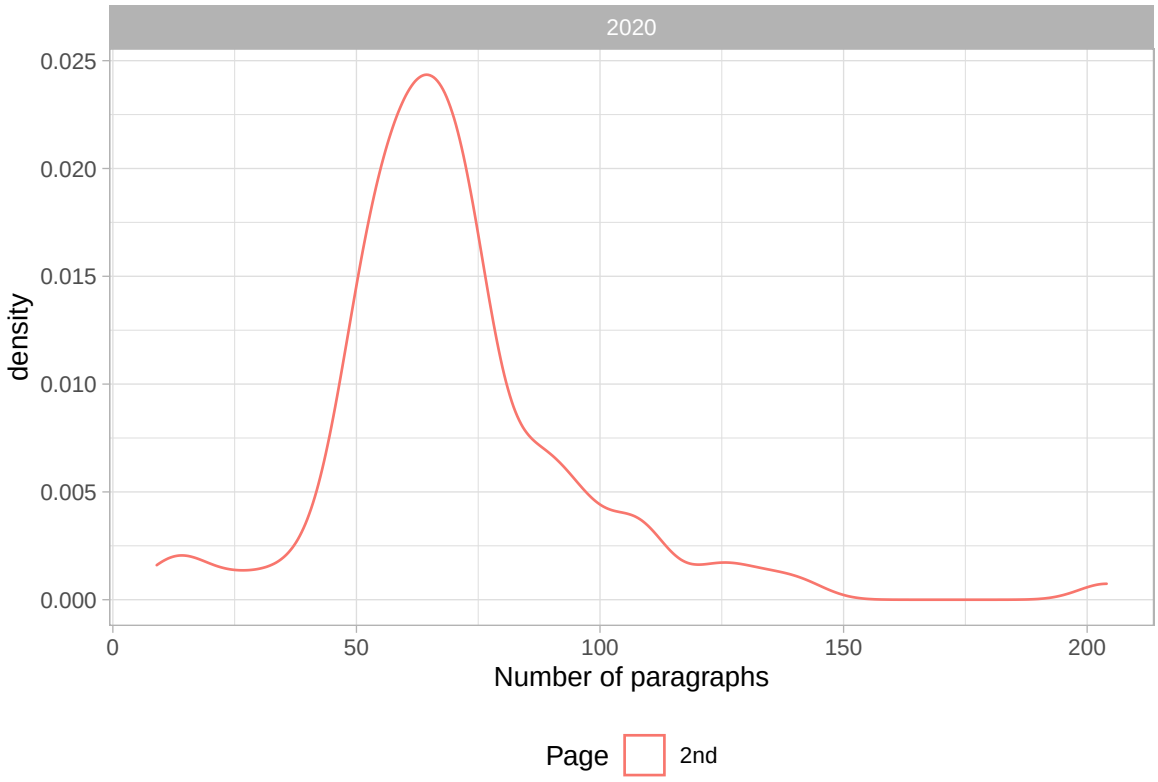


Figure 3: Distribution of Number of Paragraphs in the Newspaper Page

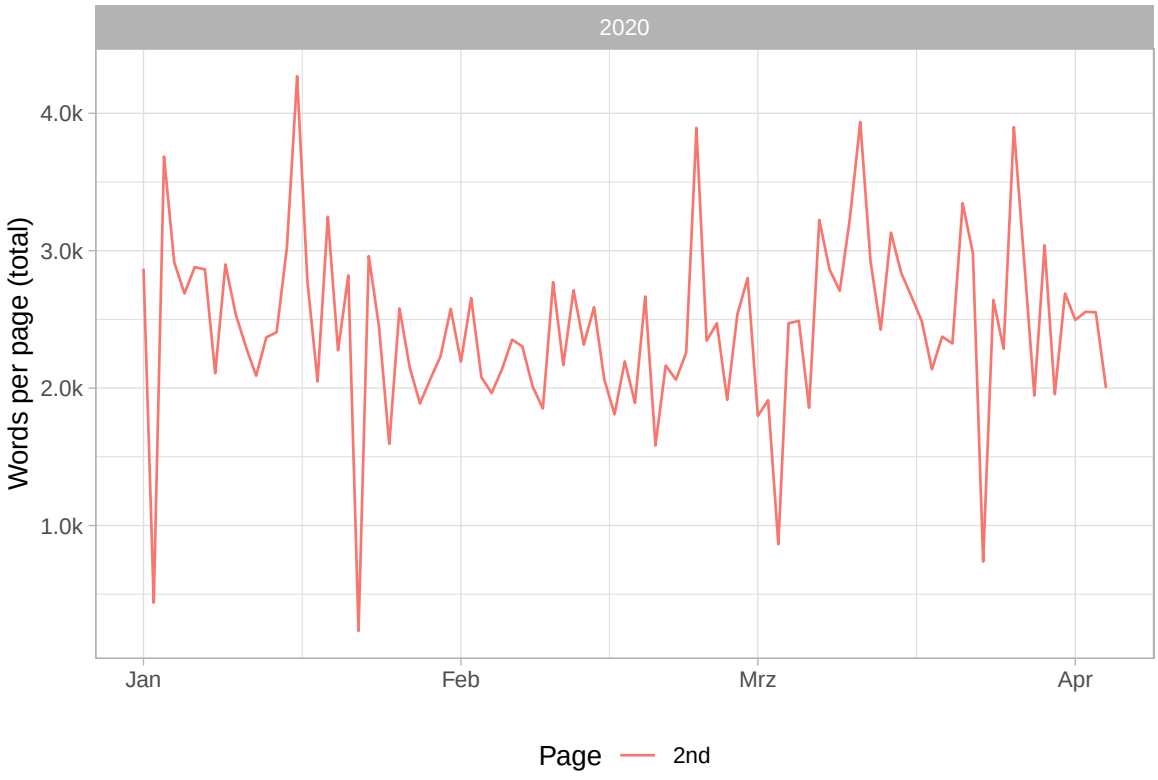


Figure 4: Word counts in the page of the newspaper per day

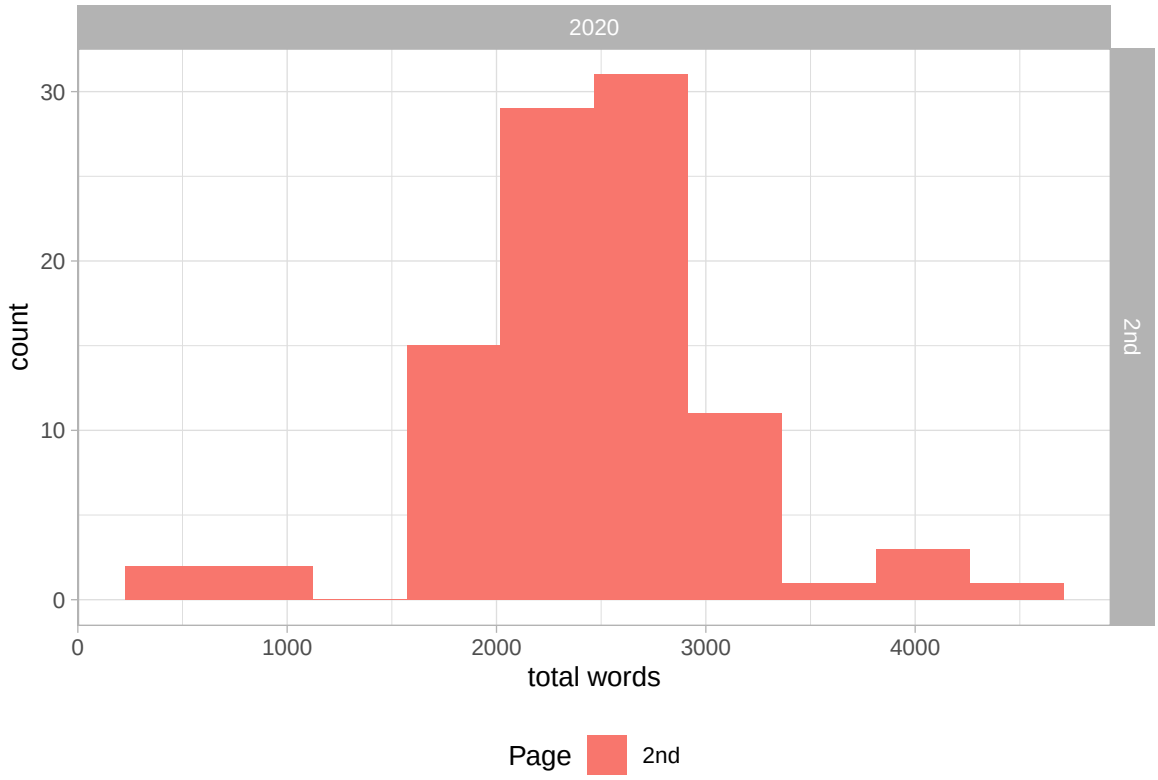


Figure 5: How bulky is a page of a newspaper in terms of word counts?

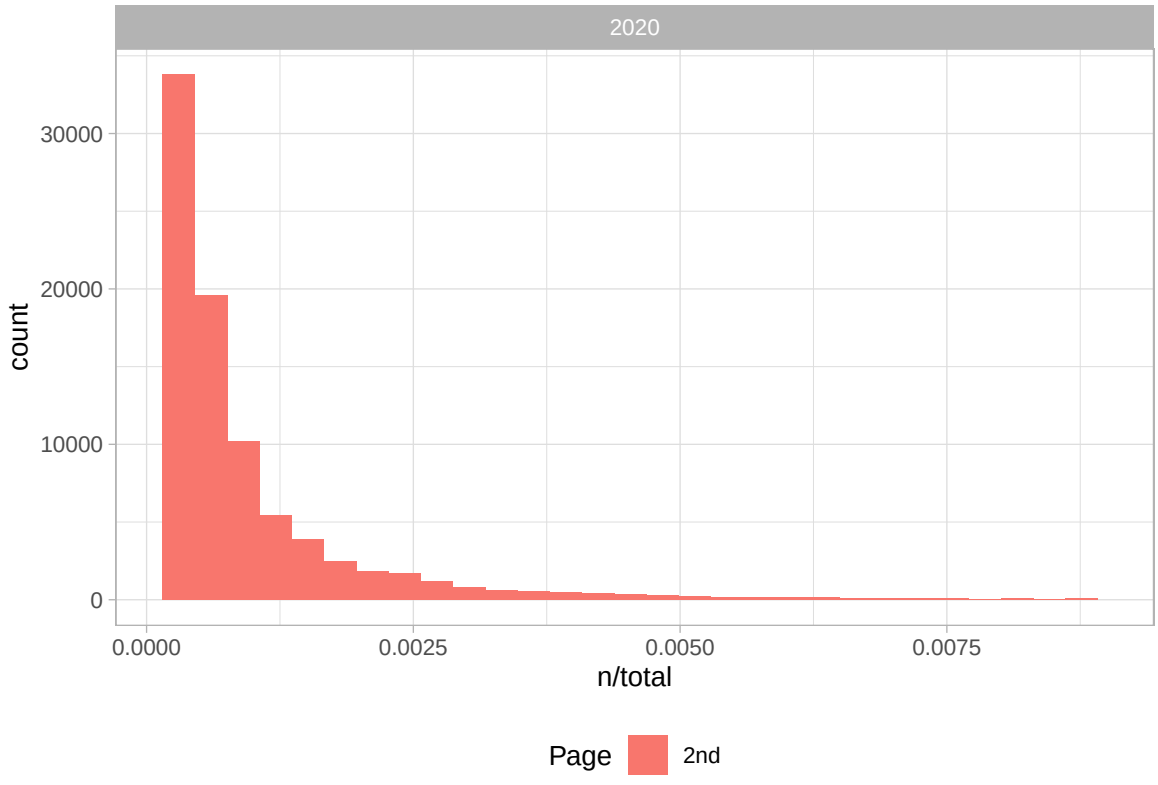


Figure 6: Term Frequency Distribution per page of the newpaper

\begin{table}[!h]

\caption{The 12 bigrams with the highest tf_idf in 2020}

page_num	month	bigram	n	tf	idf	tf_idf
2nd	Feb	outbreak prevention	468	0.0122986	1.098612	0.0135114
2nd	Jan	xi jinping	413	0.0094017	1.098612	0.0103289
2nd	Apr	<U+57F9> <U+7389>	17	0.0030292	2.484907	0.0075273
2nd	Apr	<U+8D75> <U+57F9>	17	0.0030292	2.484907	0.0075273
2nd	Apr	production complex	34	0.0060584	1.098612	0.0066559
2nd	Mrz	crown pneumonia	222	0.0047442	1.386294	0.0065769
2nd	Jan	thousand nineteen	178	0.0040521	1.386294	0.0056174
2nd	Mrz	poverty alleviation	169	0.0036116	1.386294	0.0050067
2nd	Feb	medical team	157	0.0041258	1.098612	0.0045327
2nd	Feb	medical staff	136	0.0035740	1.098612	0.0039264
2nd	Jan	central committee	132	0.0030049	1.098612	0.0033012
2nd	Mrz	political party	74	0.0015814	1.791759	0.0028335

\end{table}

\begin{table}

\caption{The 12 trigrams with the highest tf_idf in 2020}

page_num	month	trigram	n	tf	idf	tf_idf
2nd	Apr	<U+8D75> <U+57F9> <U+7389>	17	0.0049941	2.484907	0.0124099
2nd	Apr	resume production complex	33	0.0096945	1.098612	0.0106505
2nd	Apr	<U+4F55> <U+9999> <U+4E91>	12	0.0035253	2.484907	0.0087600
2nd	Feb	health care professionals	134	0.0060710	1.098612	0.0066697
2nd	Feb	traditional chinese medicine	87	0.0039416	1.386294	0.0054643
2nd	Jan	party central committee	99	0.0038165	1.098612	0.0041929
2nd	Jan	xi jinping president	99	0.0038165	1.098612	0.0041929
2nd	Mrz	strict political party	64	0.0022609	1.791759	0.0040510
2nd	Mrz	comprehensive strict political	63	0.0022256	1.791759	0.0039877
2nd	Jan	china features socialist	87	0.0033539	1.098612	0.0036846
2nd	Feb	defense joint control	65	0.0029449	1.098612	0.0032353
2nd	Mrz	rural community workers	33	0.0011658	2.484907	0.0028969

\end{table}

4.1 Word frequency time series and economic data

In order to gain an impression of the impact the frequency of a single word can have on the economy, we created a function to generate time series and visualize the results.

The user first specifies whether he wants to search the articles on page 1 or page 2 (page_num) for a specific word (eng_word). In addition, the user can specify a time period in which the analysis is to be performed using start_date and end_date. In the parameter econ_data he also determines whether he wants to compare the word frequency of eng_word with the NASDAQ_CNY or dollar_yuan_exch, which have already been explained in the section on economic data. We have integrated a function to download them from different sources and then adjust and, if necessary, normalize them according to the user's input time parameters. Furthermore there is the possibility to extend the frequency of time series from monthly to daily. It is also possible to extend the frequency of time series from monthly to daily.

In the next step we load all articles of page 1 or 2 according to the number of pages requested by the user and tokenize them. A token can be described as a meaningful word, which is worth considering in the context of a later word frequency analysis. The function which performs this task is called *tidy_text()*. Now we have a database which can be filtered by the user's chosen word eng_word, and the time period. Afterwards the values of the economic data are passed over the filtered database and the two time series are plotted. This plot is now transferred to the app and displayed to the user.

4.2 Exploratory statistics, visualisation, and text mining

The goal for this project was to show that it is possible to create an interactive app with live data updates, analysis and output. While this is just an illustration to prove the concept, there are of course limitless possibilites to extend the dataset with more economic and text data. And all manner of econometric, natural language processing, machine learning or corpus linguistig tools could be implemented.

In the app as it is now, we managed to implement the simplest descriptive statistics: - An overview of the dataset that indicates time and number of articles for page 1 and 2 respectively. - Plots of the frequency of articles per day. - The difference between the local data and the newest article online, and - plots of the frequency of certain keywords per day standardized around the starting date, in comparison to a NASDAQ index of Chinese industrial companies, also standardized the same way.

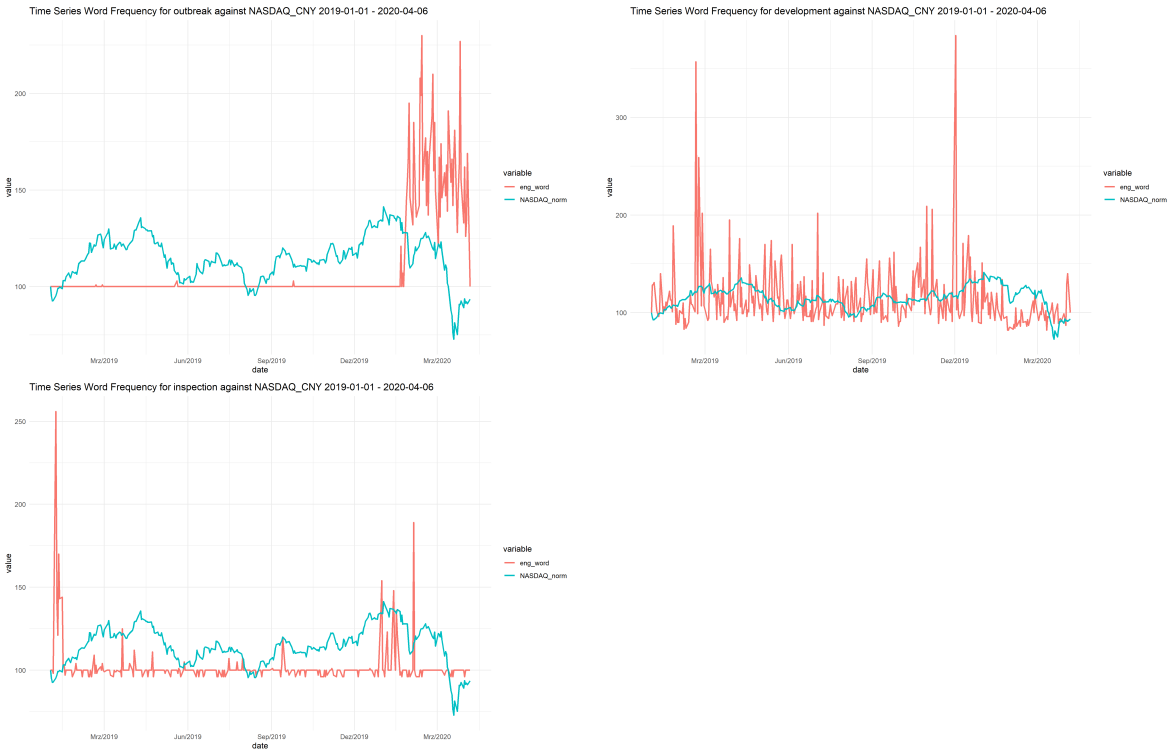
These serve to give the user an overview and a little more understand of the data. The article frequency per day plots show, for example, that there are systematic differences between the first and second pages. For both 2019 and 2020, the first page has several outliers with over 10 articles per page, while the second page has no more than 9 articles per page. This is mostly due to the coverage over important events which are covered in many small articles on the front page, e.g Xi Jinping's meetings

with world leaders at international summits.

But especially the time series shows the advantage of quantitative text analysis: Now we can see for example the frequency of “Crown”, the literal translation of Corona, or equivalently the term “outbreak”, increasing dramatically around February as expected, coinciding with a drop in the industrial index.

Another interesting result is the frequency of “development”, a staple of positive economic reporting and policy reports. While always present, especially during upturns in the index, starting in February 2020 it get’s remarkably displaced, most likely by reporting on the pandemic.

Lastly this quick and large scale analysis can lead to surprising results that warrant furher investigation: The term “inspection”, usually associated with political control and anti-corruption activities, has two extreme modes in the data, one during the outbreak and one last spring. This could be related to the spring festival, which is around February every year, or have some other reason. The results show that this approach has potential to be a highly useful tool for empirically testing hypothesis and developing new ones.



4.2.1 Interesting extensions

More classical text analysis output would have to include of course the most common words per article. This is a straighforward way of gaining a better understanding of a corpus. Even more interesting is comparing different parts of the same corpus, e.g. at

different time points. This would show systematic deviations in use of vocabulary, e.g. before and after the crisis.

Correlations are another way to reveal underlying structures of the text. Which two- or n-word groups are particularly common? How does their use vary over time? R packages including plots for these so-called “n-grams” include the aptly named “ngram”,“JSTORr”, “NSP” and “WordStat”.

4.2.2 Advanced options

On the more technical and advanced side, lda/topic models can offer great insights, but are also more difficult to interpret. Sentiment analysis is another complex, but controversial topic, which could be implemented.

4.3 Conclusion

While the connection to websites and API is inherently unstable, and the integration of a web application with different transformation and analysis tools was very challenging, we see the potential this method has in bringing approaches from text and data analysis to economics and making insights from different fields approachable to users without technical experience.

The critical component of this project was cooperation. While the use of github was of great convience, except for some merging issues, using voice chat was also useful for holding remote meetings. The project could have benefited from more consistent planning of the individual steps, which will become easier with more experience. This could have reduced some time spent merging and doubling work. The diverse tasks however gave us the opportunity to try and evaluate the usefulness of many different approaches and packages.

The complexity of ongoing issues like the Corona pandemic illustrate that researchers will benefit from contrasting different approaches and the feedback that is only possible through producing clearly documented and accessible results.

We would like to think that our project is a small proof of concept, showing what could potentially be a very useful research and publication method.

Eidesstattliche Versicherung

Ich versichere an Eides statt durch meine Unterschrift, dass ich die vorstehende Arbeit selbständig und ohne fremde Hilfe angefertigt und alle Stellen, die ich wörtlich oder annähernd wörtlich aus Veröffentlichungen entnommen habe, als solche kenntlich gemacht habe, mich auch keiner anderen als der angegebenen Literatur oder sonstiger Hilfsmittel bedient habe. Die Arbeit hat in dieser oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen.

Essen, den _____ _____

David Schulze, Eyayaw Teka Beze, Nils Paffen