38 Burstier Events

Analysing Human Memory over a Century of Events Using *The New York Times*

Joseph L. Austerweil, Charlie Pilgrim and Kesong Cao

Over the last few decades, psychologists have increasingly found that the mind stores and uses the statistics of its environment. However, less work has analysed whether the environmental statistics have changed and what that would imply for the mind. In this chapter, we consider human memory as the solution to the computational problem of predicting what events will happen next given a history of past events. Prior work examining two years of data (1986–1987) found that the environmental statistics of events occurring in the world are reflected in human memory of events, such as practice and retention effects. We analyse the last century of event statistics by assuming that words in the headlines of *The New York Times* are each an event. While presenting our methods, we do so in the form of a case study – we discuss general practices for Behavioural Data Science projects, standard issues that arise and how to resolve different issues as they arise for the presented analyses. After replicating prior work analysing event statistics in this manner during 1986–1987, we extend the methodology to the last century (1919–2019). Our analyses suggest that the events are occurring in denser bursts, meaning that, if a new event occurs in the last few years, this event reoccurs more often in the short-term and less often in the long-term (as compared to events that first occurred in the early twentieth century). This suggests that human memory faces different environmental demands than it has in the past and may be adapting to the dynamics of event statistics.

Keywords

behavioural data science, rational analysis, human memory, event prediction, longitudinal data analysis.

38.1 Introduction

Over the last century, technology and globalisation have increased the pace of everyday life (e.g., Kuzemko et al., 2025). Within the last two decades alone there have been two worldwide recessions, a pandemic, multiple wars and many other important events. Given that natural event dynamics have likely changed, the task demands on human memory and prediction have likely also

changed. As environmental information has increased over the last century (Hills, 2019), has the pace at which events are reported to occur also changed over the same period? What implications would this have for human memory? By analysing large data sets of human behaviour, we can investigate these questions in a manner that would have been impossible to do in the past.

Predicting events and acting accordingly are basic adaptive functions that rely heavily on human memory. One might expect adaptive behaviour to tend towards the behaviour that a rational agent would exhibit in the same situation – given the same environment, goals and limitations. As such, researchers often use rational analysis (Anderson & Milson, 1989; Anderson, 1990) to test whether human memory serves prediction in an optimal manner. From this perspective, researchers explain the nature of cognitive faculties by appealing to the relevant environmental statistics, rather than focusing on mechanisms. Across diverse domains, such as visual perception, language learning and memory, researchers have found that the mind has internalised the statistics of its environment and further that it can do so in the course of a laboratory experiment (Anderson & Schooler, 1991; Saffran et al., 1996; Fiser & Aslin, 2002; Geisler, 2008; Romberg & Saffran, 2010). Although this adaptationist perspective is not without its issues (see e.g., Gould & Lewontin, 1979; Jones & Love, 2011), it provides a unified framework for understanding and investigating disparate aspects of human cognition. It has been a useful tool for understanding disparate phenomena related to human memory. Anderson and Milson (1989) showed that the dynamics of memory (recency, frequency and spacing of practice) can be understood as optimising event prediction, given the environmental statistics of events. For example, human memory for an event and the probability of an event being reported on again in natural language both decay as power–law functions over time. Following up on this work, Anderson and Schooler (1991) validated these statistical assumptions by conducting analyses of written text reporting world events (e.g., words in headlines from *The New York Times*).

In this chapter, we investigate human memory by studying the information environment that human cognition is operating within (i.e., the problems faced by human memory due to its environment). We ask the following questions: What are the environmental statistics of events? Have these statistics remained stable or changed over the course of human history? To address these questions, we examine the dynamics of event statistics over the last century. We do so by utilising a large longitudinal data set of human events: Headlines of *The New York Times*. This showcases the power of behavioural data analysis as it can be used to investigate questions that would be impossible to explore using standard laboratory, experience sampling or most other methodologies.

The outline of this chapter is as follows. First, we review prior work on human memory that uses ecological statistics to understand memory retrieval. Second, we describe how raw data are gathered from online sources and how to transform the data into a form that is usable for analysis. We cover some problems that arise during data cleaning, and how to solve those problems.

Third, we conduct an analysis of events in headlines from *The New York Times*. We do so by first validating our methods by reproducing a prior analysis of headlines from the front page of *The New York Times* from 1986 to 1987 (Anderson & Schooler, 1991). Lastly, we extend these analyses to the last century of headlines (including those not present on the front page) and discuss their implications with respect to how task demands for memory retrieval have changed over the last century.

38.2 Information Retrieval, Human Memory and the Ecology of Events

How does human memory retrieve events relevant to the current context? Anderson and Milson (1989) presented the first rational analysis of human memory retrieval. The general principle is that an optimal memory structure should mirror the environmental structure, so that memories are retrieved in order of their probability of being needed. More specifically, they formulated the problem of human memory retrieval in terms of optimal information retrieval and adapted a model created to solve an analogous problem in the library sciences (Burrell, 1985). To do so, they derived how a Bayesian agent would predict events given several reasonable assumptions regarding the psychological mechanisms of human memory retrieval and the environmental structure of events. That Bayesian model combines a prior probability based on a non-homogeneous Poisson process with a cue-based likelihood function. According to a non-homogeneous Poisson process, the probability of k events occurring in a period of time (or more generally, some region of a domain) is Poisson distributed with parameter determined via the integral of the non-homogeneous rate function over that time (for more details, see Kingman, 1992). In this model, the rate increases for some period of time whenever an event occurs. A cue-based likelihood function assigns larger probabilities when the cues based on the retrieval task and those of a given memory are consistent. Then, the model uses a noisy version of the following rule to recall items: recall items in descending order of probability that the events are likely to be needed (need probability as assessed by the Bayesian model), until the need probability falls below some threshold. We refer a reader interested in the details of the model derivation to prior literature (Anderson & Milson, 1989; Anderson, 1990; Schooler & Anderson, 1997).

Given first principles of the computational task and environmental statistics of events, Anderson and Milson (1989)'s model captures challenging phenomena in human memory performance. We will examine two of these phenomena in this chapter:

- *Frequency effect*: How does the frequency of word occurrences in the past 100 days predict the probability of the word occurring on the 101st day? This is analogous to the practice effect in human memory.

- *Recency effect*: How does the number of days since a word occurred predict the probability of it appearing on the next day? This is analogous to the retention effect in human memory.

The recency effect is that memory performance is better when closer in time to its last observation. The particular functional form for how performance decays over time observed in laboratory experiments is power-law. To produce this effect, Anderson and Milson (1989)'s model relies on assuming that humans observe events that share this power-law form. Anderson and Schooler (1991) confirmed this environmental assumption of event statistics needed by the model to reproduce human memory phenomena by analysing two years of headlines from the front page of *The New York Times*, a database of child directed speech (CHILDES; MacWhinney & Snow, 1990) and almost five years' worth of electronic mail sent to John R. Anderson. Note that, although we adopt this Bayesian model to motivate for our analyses, this should not be viewed as an endorsement of a particular approach for modelling human memory. To us, each computational framework has complementary strengths and weaknesses when used to create models of human behaviour. We take a Bayesian perspective in this work as it is (from the authors' perspective) the most straightforward for pursuing analyses of environmental statistics for any agent with a memory system. We refer readers interested in a Connectionist model that can capture similar memory phenomena to Mozer et al. (2009).

In this chapter, we first replicate the work of Anderson and Schooler (1991) to analyse the relationship between human memory phenomena and the statistics of events. Once we validate our methods by comparing our results to prior results, we extend the analysis to the last century of events. We focus on two psychological effects of human memory in terms of the structure of events found in headlines in *The New York Times*: the frequency and recency effects. In the rest of this section, we provide a summary of these effects and equations often used to describe human performance.

38.2.1 The Frequency/Practice Effect

The more an item is observed by a person, the more likely it is that the person will remember that item. One classic demonstration of this effect comes from Ebbinghaus (1885) testing himself on learning sets of lists of 'made-up' but possible words (nonsense word lists). By practicing nonsense word lists a different number of times (e.g., list 1 might be practiced once, list 2 might be practiced four times), he controlled the amount of exposure – the frequency – to each word list. He found that it took many fewer attempts to perfectly reproduce a list that was practiced more frequently than a word that was practiced less frequently. His performance followed a power-law function (Newell & Rosenbloom, 1981; Anderson & Schooler, 1991):

$$Attempts = aS^{-b}, \tag{38.1}$$

where a and b are free parameters (held constant across different words) and S is the amount of practice for that item. Often, power-law functions are plotted in log-log space (where the x-axis is $\log(x)$ and the y-axis is $\log(y)$). This is because they are linear when plotted this way (although some caution is needed in using this logic as it is not the only function that can appear linear when plotted in log space – see Mitzenmacher, 2004). Anderson and Schooler (1991) found that Ebbinghaus (1885)'s performance as a function of practice was best fit as log Attempts $= 6.24 - 0.124 \log S$, where S is the number of days of practice. Intuitively, $b = -0.124$ is Ebbinghaus' memory learning rate.

38.2.2 The Recency/Retention Effect

Following the observation of an event, the probability that a person will remember that event decays over time (Ebbinghaus, 1885; Wickelgren, 1976). A natural first assumption for the pattern of decay would be the exponential function, as in radioactive decay. Experimental evidence points instead to a power law function (Ebbinghaus, 1885; Anderson & Schooler, 1991). A power law is similar to an exponential function. The main practical difference for this context is that it allows that a few events are still remembered long after their last occurrence (power law functions decay more slowly than exponential functions; Simon, 1955; Mitzenmacher, 2004). The probability that a person retrieves a memory after time T from its last occurrence (or other performance measures for memory) is

$$Performance = cT^{-d}, \tag{38.2}$$

where c and d are free parameters. As in the practice effect, Ebbinghaus (1885) investigated this phenomena through learning of nonsense word lists. He measured how easy it was to re-learn a nonsense word list after some time, in terms of per cent savings from learning a list never before seen. Anderson and Schooler (1991) found that Ebbinghaus (1885)'s retention performance was best fit as log Per cent Savings $= 3.862 - 0.126 \log T$, where T is the delay length in log hours.

Intuitively, $b = -0.126$ is Ebbinghaus' forgetting rate.

Beyond the practice and retention effects, there are other dynamics that have been observed in human memory, which can be captured within this framework. For example, the spacing effect refers to how the distribution of event occurrences over time affects memory performance. If someone observes an event three times over an hour, does it matter whether those three occurrences all happen within the last five minutes of the hour or distributed about evenly over the hour (e.g., 1 minute, 30 minutes and 59 minutes). This and the many effects found by memory researchers are beyond the scope of the current analyses. To the best of our knowledge, conducting this style of analysis for the spacing effect remains an open problem (although large-scale laboratory and real-world studies have replicated this effect; Kim et al., 2019). In Section 38.3, we will describe our methodology for replicating Anderson and Schooler

(1991) by investigating whether the frequency and recency of events in headlines of *The New York Times* reflect the power law distributions seen in the practice and retention effects in human memory. We will then extend this by analysing the effects in each of the last 100 years, to ask whether the environmental structure is stable or changing over time.

38.3 Methods

As people go about their lives they produce, passively (e.g., GPS coordinates of their location over time) and actively (e.g., a post on Twitter), large amounts of data that are recorded for later access. Over the last decade researchers have taken advantage of large naturally occurring data sets of human behaviour as a means for analysing psychological mechanisms, theories and phenomena in a more naturalistic and large-scale manner than is feasible in traditional laboratories (Harlow & Oswald, 2016; Paxton & Griffiths, 2017). For example, Hopman et al. (2018) explored how different word-level factors influence second language learning by analysing a large, anonymised data set of user performance from a second language learning application (Duolingo). Further, Pilgrim et al. (2021) analysed how increasing demands on human attention may be driving historical changes in language use in the Corpus of Historical American English (Davies, 2010). In our case we are interested in the text of newspaper headlines and their publication dates.

Ideally, we would use all headlines within our desired date range. There are some common methods to gather this kind of data: downloading a released data set, web crawling or using an Application Programming Interface (API) provided by a data owner. An API provides access for developers and researchers to request and download portions of their data, usually through web requests (Table 38.1 shows some examples). APIs are often used by corporations and other large organisations to control what data and how much of it can be

Table 38.1 *Some example big data sources. Official APIs usually require an authorisation key to access. Third party data sources are often easier to access but can have a lower quality or quantity of data*

Source	Type	Link
The New York Times	official	https://developer.nytimes.com/apis
Reddit	official	www.reddit.com/dev/api
	third-party	https://github.com/pushshift/api
Twitter (currently, X)	official	https://developer.twitter.com/en/docs/twitter-api
	third-party	https://github.com/twintproject/twint
GitHub	official	https://docs.github.com/en/rest/reference/activity
	third-party	www.gharchive.org

accessed by third-parties. It is worth noting that data from an API is often generic, containing more fields than what a researcher needs, and almost always in a different format than what we desire. Therefore, data cleaning and pre-processing (sometimes called data wrangling) are especially important when gathering data through an API.

In this chapter, we are exploring headlines from *The New York Times*, which provides APIs on their developer website (https://developer.nytimes.com/apis). We are mainly interested in their Archive API (https://developer.nytimes.com/docs/archive-product/1/overview), which allows us to download articles specifying the year and month. To start downloading, we first need to request an API key by registering an account. This is common when using APIs to gather data from a corporation as it allows them to monitor your data requests via your unique key. Once we have an API key we can request data by accessing the API endpoint URL (Uniform Resource Locator, web address) while specifying the year and month following strict formatting rules: 'https://api.nytimes.com/svc/archive/v1/{year}/{month}.json?api-key={your_key}'. For example, here the 'year' is the usual YYYY format (e.g., 2021) and the 'month' is without a leading zero (e.g., 1, not 01).

To gather data requiring a large number of data requests (e.g., all headlines from *The New York Times* over the last 100 years), it is practical to automate the process. Otherwise, one would need to manually modify the request for each month and year and visit the URL in a browser. We use a Python script, which is included as Listing 38.1. Of course, one can also use other programming languages such as a shell script, R or Java. It is essential to include a delay between each request as most APIs only allow a certain number of requests for a given time period (this is usually listed on the same website as where you obtain the API key).

Figure 38.1 illustrates the general flow of a typical data analysis project.

```
import time

import requests # Library for accessing web pages

def scrape_nyt_archive():
  """
  Scrapes data from the NYT Archive API
  Saves data as a json file for each month and year
  """
  start_year = 1919
```

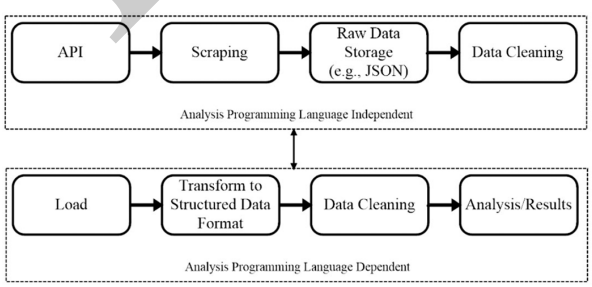

Figure 38.1 *A flowchart depicting the typical process of a data analysis research project*

Project tasks can usually be broken into two types: tasks that are independent (top dashed rectangle) and dependent (bottom dashed rectangle) of the programming language used for data analysis. It is important to include and save the raw and intermediate data in a manner that is not dependent on the programming language (e.g., JSON files). This facilitates reproducibility and finding errors in the processing pipeline, as well as enables future work (by you or other researchers) to use other programming languages for analysis. Although python and R are the most common languages for data analysis (as per the writing of this book chapter), they may not be in the future (perhaps they are not by the time you read this chapter!). For example, raw data from some early work by one author of this book chapter were stored in a Matlab-specific manner. The author later regretted that decision. The bidirectional arrow between the language-dependent and -independent tasks highlights that errors in earlier stages of data processing are often discovered while analysing the initial results.

Listing 38.1 Python script to batch download from NYTimes Archive API

```
import time

import requests  # Library for accessing web pages

def scrape_nyt_archive():
    """
    Scrapes data from the NYT Archive API
    Saves data as a JSON file for each month and year
    """
    start_year = 1919
    end_year = 2019

    # Add your own API key here for the NYT Archive
    api_key = "<ADD YOUR OWN API KEY HERE>"

    # Loop through years
     for year in range(start_year, end_year + 1):
        # Loop through months
        # This loop stops at 12, one below 13
        for month in range(1, 13):
            print("Working on", year, month)

            # Create the correct URL based on the year, month and API
            key
            api_base_url = "https://api.nytimes.com/svc/archive/
            v1/{}/{}.json?api-key={}"
            api_url = api_base_url.format(year, month, api_key)
```

```
        # Get the response from the API
        archive_response = requests.get(api_url)

        # Extract the content of the response
        archive_data = archive_response.content

        # Save the data from the NYT archive to a file
        output_filename = "../data/raw/{}-{}.json".format
        (year, month)
        with open(output_filename, 'wb') as f:
            f.write(archive_data)

        # Add a delay between two downloading requests
        # So the API doesn't ban us for using it too frequently
        time.sleep(10)
```

The Python script saves the data for each year and month into a separate JavaScript Object Notation or JSON data file. Despite having JavaScript in its name, JSON is a language-independent structured data format. It consists of a list of key-value pairs that are encapsulated in braces. Each value can be strings, numbers or full JSON object, which allows for rich structure. Keys are strings (often literal strings that are surrounded by apostrophes or quotes, such as ''key' or "key"). For example, the downloaded January 1986 raw data are structured like this:

```
file (in JSON format)
-    copyright (contains the copyright information which
     we don't need)
-    response (the data)
-    meta (contains metadata which we don't need)
-    docs (contains a list of data relating to NYT
     articles, which we need)
```

The first item in 'docs' is shown here:

```
{'web_url':  'www.nytimes.com/1986/01/01/garden/food-
notes-242086.html', 'snippet': "Talking About Drinking
Parents and young people who are concerned about reas
'lead_paragraph': "Talking About Drinking Parents and
```

```
young people who are concerned about ' abstract' : None,
' print_page' : ' 31' ,
' blog' : [ ] ,
' source' : ' The New York Times' , ' multimedia' : [ ] ,
' headline' : { ' main' : ' FOOD NOTES' } ,
' keywords' : [ { ' name' : ' subject' , ' value' : ' FOOD' } ] ,

' pub_date' : ' 1986-01-01T00:00:00Z' ,
' document_type' : ' article' , ' news_desk' : ' Living Desk' ,
' section_name' : ' Home and Garden' , ' subsection_name' :
None,
' byline' : { ' person' : [ { ' firstname' : ' Nancy' , ' middle-
name' : ' Harmon' ,
' lastname' : ' Jenkins' , ' rank' : 1,
' role' : ' reported' ,
' organization' : '' } ] ,
' original' : ' By Nancy Harmon Jenkins' } , ' type_of_mater-
ial' : ' News' ,
' _id' : ' 4fd148da8eb7c8105d625847' , ' word_count' : 977,
' slideshow_credits' : None}
```

We can see that the headline text is at doc['headline']['main'] and the publication date is at doc['pub_date']. To get only front page headlines, we can use the field doc['print_page'] as a filter. Note that most fields are simple to decipher based on their names. There is no complete documentation on the NYTimes developer website (as of 9 August 2021). This is common for online datasets. If it is not easy to decipher a field name, one strategy for deciphering it is to display all unique values of a field and guess based on them. This is generally good practice as there are often unexpected values or mistakes in large data sets. For example, across the year 1986 to 1987, the doc['print_page'] field contains primarily numbers, but there are two special values None and 'D'. We also note that all numbers are in string format and we need to transform valid values to numbers if we want to interpret them in that manner. For example, this would be necessary for an analysis of the relationship between frequency and the page number of a word's occurrence.

After downloading the raw data in JSON format, the next step is to extract the relevant information, do any data processing or cleaning and save it in a more convenient data format for your own analysis. We convert the raw data to a Pandas dataframe for each year in Listing 38.2.

First, we iterate through the months of the year and open the matching JSON file. Once loaded into python, JSON can be navigated like a large python dictionary to extract the data that we need. In this case we extract the date of publication and convert this into a more usable format. We also extract the

headline text and do some text cleaning, using the function process_headline_into_word_list (Listing 38.3). Finally, we convert the data for the entire year into a Pandas DataFrame. Pandas is a python data science package designed for handling and manipulating large databases. We save this DataFrame as a pickle. A pickle is a raw data format that provides efficient data storage and access but is specific to the programming language that it was created in. This will allow us to access the NYTimes data easily later.

Listing 38.2 Building a DataFrame out of the raw data

```
import json
import datetime
import pandas as pd

def process_nyt_json_into_pandas_df(year):
    """
    Extracts the NYT raw headline data from JSON files for a year.
    Processes the headlines to clean the data and extract words.
    """
    headline_full_data = []  # Save the cleaned data here

    # Iterate through months
    for month in range(1, 13):
        # The filepath will depend on where the raw data was saved
        json_filepath = "../data/raw/{}-{}.json".format(year,
        month)

        # Open the JSON file for this year and month
        with open(json_filepath) as json_file:
            # Extract the data we want, i.e., "docs"
            json_data = json.load(json_file)
            docs_data = json_data["response"]["docs"]

            # For each document - extract, clean, and save the data
            for doc in docs_data:
                # Get the first 10 characters "1986-01-01" from "1986-
                01-01T00:00:00Z"
                date_string = doc["pub_date"][:10]

                # Extract the year, month, and date
                doc_year = int(date_string[:4])
                doc_month = int(date_string[5:7])
                doc_day = int(date_string[8:])
```

```
            # Use datetime to get the day of the year as a day_
            index
            date_val = datetime.date(year=doc_year, month=
            doc_month, day=doc_day)
            day_index = int(date_val.strftime('%j'))  # The
            day of the year as a number

            # If the headline is in the expected format, extract
            the headline text
            if type(doc["headline"]) == dict and "main" in doc
            ["headline"]:
                headline_words = process_headline_into_word_
                list(doc["headline"]["main"])

                # Save the cleaned data along with the date of the
                headline
                headline_full_data.append(
                    [headline_words, day_index, doc_year, doc_
                    month, doc_day, doc]
                )

# Convert the data into a pandas DataFrame and save it
headlines_df = pd.DataFrame(
    headline_full_data,
    columns =["headline", "day_index", "year", "month",
    "day", "doc"]
)
output_filepath = "../data/clean/clean_headlines_{}.pkl".
format(year)
headlines_df.to_pickle(output_filepath)
```


We created a separate function to handle the text cleaning (Listing 38.3). In order to process the headlines, we used regular expressions and a powerful text manipulation tool. First, we converted the text into ASCII characters only, which could prevent future complications. Then, we removed most non-alphanumeric characters, except spaces and the single quote character (‘). Following that, we converted all letters to lower case, as the original headline texts contain many all-capitalised and first-letter-capitalised words. We realise this method will have the limitation of confounding homographs or words that share the same letters but have different meanings (e.g., ‘AIDS’ and ‘aids’). These issues can be ameliorated using natural language processing methods provided by popular packages such as NLTK and StanfordNLP. The last text manipulation step we conducted was to remove the starting and ending single

quotes in case it was split within a quote (e.g., "'A B'" would otherwise be transformed to "'A" and "B'", rather than "A" and "B"). Finally, we removed stopwords, which are common words such as prepositions, pronouns and articles, for example, 'of', 'he', 'a'. These are not closely related to specific events and so are not relevant to our analysis.

Listing 38.3 Cleaning raw headline text into a list of words

```
import re

# Import stopwords from a txt file as a word list
# Txt file is downloaded from a Stanford library for Natural
Language Processing
#  https://github.com/stanfordnlp/CoreNLP/blob/main/data/edu/
stanford/nlp/patterns/surface/stopwords.txt
STOPWORDS = open("resources/stopwords.txt", "r").read().spli-
tlines()

def process_headline_into_word_list(headline):
    """
    Take a headline as a string input.
    Clean the headline to standardise the characters.
    Return the cleaned headline as a list of words.
    """

    # Process the headline string to remove or transform unwanted
    characters
    # Ignore non-ASCII characters
    s = headline.encode("ascii", errors="ignore").decode()

    # Remove non-alphanumeric characters, except spaces and single
    quotes
    s = re.sub(r"[^\w\s\']+", "", s)

    # Convert to lowercase
    s = s.lower()

    # Split the headline string into a word list
    word_list = s.split(" ")

    # Start a new list of cleaned words
    clean_words = []
    for w in word_list:
        # Remove starting and ending single quotes
```

```
        w = re.sub(r"^' +", "", w)
        w = re.sub(r"' +$", "", w)

        # Do not include the word if it is a stopword
        if w.lower() not in STOPWORDS:
            # Add the cleaned word to the clean word list
            clean_words.append(w)

    return clean_words
```

Another popular method for cleaning text data is stemming or lemmatisation (see Schütze et al., 2008). Many words have different forms, known as morphologies, that can modify the meaning, such as house, housing, houses, housed. Stemming and lemmatisation are methods to group these words together as belonging to the same lemma. In the context of human memory, one could argue that modified word forms relate to events that are associated with the same lemma, for example, house, and so this could be a suitable step in data processing for this study. Conversely, one could argue that modified word forms do convey different meanings and lemmatisation brings in extra assumptions that reduce the information contained within the language, and we should instead follow the natural structure of words. In our case, we decided to follow the latter argument and not to apply this method, in part to be consistent with the prior work by Anderson and Schooler (1991).

38.4 Analysing a Century of Events

Once the data are cleaned in a Pandas DataFrame object, we can begin analysing the frequency and recency of effects. As a first step, we replicate Anderson and Schooler (1991)'s analysis using comparable headlines from *The New York Times* that were gathered through the API. This serves as a validation of the methodology (and an error check), as well as verifying the robustness of results from prior work. With the methodology validated, we can examine frequency and recency statistics for each year across a century of headlines. In their analysis, Anderson and Schooler (1991) considered headlines from the two year period 1986–1987. We replicated the analysis for that two-year period. However, for the trend analysis we will use one-year periods, as this is more natural when looking at trends over a century. The code listings here are for one-year periods.

Anderson and Schooler (1991) investigated frequency and recency effects in terms of daily occurrences of words in headlines. They assumed that each word in a headline constitutes the occurrence of an event. Although we considered other possible definitions for an event (e.g., the main noun phrase of the headline), we ultimately decided to use the same definition for an event.

Table 38.2 *Some example data from the cleaned DataFrame. The 'headline' column contains a cleaned word list extracted from the JSON doc["headline"]["main"] field*

year	day_index	doc["headline"]["main"]	headline
1921	155	Harvester Cuts Its Dividend	[harvester, cuts, dividend]
1943	358	Court Backs OPA Turkey Price	[court, backs, opa, turkey, price]
1969	12	FULBRIGHT ASSURES SPAIN ON CRITICISM	[fulbright, assures, spain, criticism]
1986	155	PACT ON EXTRADITION IS GAINING	[pact, extradition, gaining]
1993	195	School Reform Gridlock	[school, reform, gridlock]
2005	50	Yogi gets 100% emotional, and Then Some, for a Pal	[yogi, gets, 100, emotional, pal]

We did so to maintain consistency with prior work and because there was no other choice that was obviously better to us. However, this is a potential limitation and future work should examine whether other assumptions produce contradictory results.

Defining an event to be a word determines the level of granularity that we need to maintain for calculating the statistics of interest. In our case, we need to know the specific days that each word occurred. For each year we built a Python dictionary object with each word as the keys and the values being a list of the days that the word occurred. The Python code to calculate it is shown as Listing 38.4 and uses the following algorithm: First, we load in the cleaned data set, which is in DataFrame format (Table 38.2 shows some example rows).

We iterate through each row of the cleaned DataFrame, where each row represents a headline with columns including the headline text as a cleaned word list and the day of the year that the headline occurred. At each row, we loop through the word list and add the days that the word occurred to the word occurrences Python dictionary. This new dictionary data format will be the basis of calculating the statistics for the frequency and recency effects.

Listing 38.4 Building the word occurrences dictionary for a specific year

```
import os
import pandas as pd

def get_word_occurrences(year=1986):
    """
    Load the cleaned NYT headlines data for a specified year.
    Create a dictionary recording the day of year each word
    appeared.
    """
```

```
# Get the location of the cleaned data and load it
path_to_clean_data_file = "../data/clean/"  # Where you stored
the cleaned data
clean_filename = "clean_headlines_{}_with_stopwords.pkl".
format(year)
clean_filepath = os.path.join(path_to_clean_data_file, clean_
filename)
df = pd.read_pickle(clean_filepath)

# Initialize the dictionary to store word occurrences
all_words_to_day_dict = {}

# Each row is a headline with the text and day it occurred
for index, row in df.iterrows():
    # row.headline is a list of the words in the headline
    for word in row.headline:
        # Record the day each word occurred in the dictionary
        if word in all_words_to_day_dict:
            # If the word is already in the dictionary, append
            this day_index
            all_words_to_day_dict[word].append(row.day_index)
        else:
            # If the word isn't already in the dictionary, add it
            with this day_index
            all_words_to_day_dict[word] = [row.day_index]

return all_words_to_day_dict
```


For word frequency, Anderson and Schooler (1991) calculated the probability of a word occurring in the next day's headlines (they chose the 101st day) given its prior occurrences in the past X days (they chose $X = 100$ days). From a statistical point of view, we are interested in estimating the probability that a word occurs on a day given it occurred f times in the previous 100 days window. We want to use all of the data available to us so we focus not only on the 101st day of each year, but all days from 101 to 365 (we start at 101 so that we have a big enough window of 100 days to look back on). A Python script to estimate these probabilities as a function of f is shown in Listing 38.5. The function takes as an input a dictionary of word occurrences for a year as calculated in Listing 38.4. We create two lists of length f in which we count the number of times a word either occurs (freq_data_pos) or does not occur (freq_data_neg) given it occurred f times in the previous 100 days, where f is the index of those lists. We need the number of times the word does

not occur to calculate a probability, as it is the number of times a word occurred divided by the sum of the number of times it did occur and the number of times it did not occur.

Listing 38.5 Building the probability data on word frequency

```
import numpy as np

def get_frequency_prob_data(all_words_to_day_dict):
    """
    Given a dictionary mapping words to lists of days they occurred,
    estimate the probability distribution of word occurrences
    based on prior appearance frequency within a window of days.
    """

    window_size = 100
    freq_data_pos = np.zeros(window_size + 1)
    freq_data_neg = np.zeros(window_size + 1)

    for word, occurrences in all_words_to_day_dict.items():
        # Get unique occurrences as a numpy array
        occurrences = np.array(list(set(occurrences)))

        # Test every day between day 100 and day 365
        for test_day in range(window_size, 366):
            # Count the number of occurrences in the window
            window_begin = test_day - window_size
            num_prior_occurrences = np.count_nonzero(
                (occurrences >= window_begin) & (occurrences <
                test_day)
            )

            # Check if the word appeared on the test_day
            if test_day in occurrences:
                # If the word appeared on the test day, add to the
                positive count
                freq_data_pos[ num_prior_occurrences] += 1
            else:
                # Otherwise, add to the negative count
                freq_data_neg[ num_prior_occurrences] += 1

    # Estimate probabilities
    freq_data_prob = freq_data_pos / (freq_data_pos + freq_data_neg)

    return freq_data_prob
```

We then iterate through each of the words and each of the test days, counting the number of occurrences in the last 100 days, f, and then counting the word in the appropriate list depending on whether it occurred on the test day. Finally, we estimate the probability of a word occurring given f as a maximum likelihood estimate by dividing the number of times a word did occur after appearing f times over the total number of instances that a word occurred f times before a test day. For word recency, Anderson and Schooler (1991) calculated the probability of a word occurring on a given day (they chose the 101st day) given the number of days since its last occurrence. We also computed these data using the code presented as Listing 38.6. This uses a similar algorithmic structure as when calculating the frequency statistics. Specifically, for each word, we iterate through test days $x(100 \leq x \leq 365)$ and then count how many days passed since its previous occurrence $x'(x - 100 \leq x' < x)$. This process gives us a maximum likelihood estimate of the probability of a word occurring given x days since its last occurrence.

Listing 38.6 Building the probability data on word recency

```
def get_recency_data_prob(all_words_to_day_dict):
    """
    Given a dictionary mapping words to lists of days they occurred,
    estimate the probability distribution of word occurrences
    based on recency
    (how recently the word last appeared before a given day).
    """

    window_size = 100
    recency_data_pos = np.zeros(window_size + 1)
    recency_data_neg = np.zeros(window_size + 1)

    for word, occurrences in all_words_to_day_dict.items():
        # Test every day between day 100 and 365
        for test_day in range(window_size, 365):
            # Get occurrences as a numpy array
            occurrences = np.array(occurrences)

            # Get all occurrences before the test_day
            prior_occurrences = occurrences[ occurrences < test_day]

            # If the word has not appeared before test_day, skip
            if len(prior_occurrences) == 0:
                continue  # Move to next test_day

            # Work out how many days it has been since the word occurred
```

```
        last_occurrence = np.max(prior_occurrences)
        days_since_occurrence = test_day - last_occurrence

        # If days_since_occurrence is greater than window size,
        skip
        if days_since_occurrence > window_size:
            continue  # Move to next test_day

        # Now see if the word occurred on the test_day
        if test_day in occurrences:
            # If the word appeared on the test day, add to the posi
            tive count
            recency_data_pos[ days_since_occurrence] += 1
        else:
            # Otherwise, add to the negative count
            recency_data_neg[days_since_occurrence] += 1

# Estimate probabilities from the frequencies
recency_data_prob = recency_data_pos / (recency_data_pos +
recency_data_neg)

return recency_data_prob
```

38.5 Results for 1986–1987

In order to investigate the dynamics of the information environment, we replicated Anderson and Schooler (1991)'s analysis of headlines from *The New York Times* for the two year period 1986–1987. They began by asking the simple question of what the probability is of a word appearing on day 101, given how many days it occurred in the past 100 days. This is replicated in Figure 38.2 using empirical probabilities estimated from data gathered through *The New York Times*'s API (as well as showing the results published in Anderson and Schooler 1991). As a word occurs more often in the past 100 days, its probability of occurring on the 101st day increases – qualitatively mirroring the effect of practice (where the participant's performance increases with the amount of practice). Our results replicate Anderson and Schooler (1991) quite closely, which we can analyse via the following linear regression:

$$probDay101 = frequencyIntercept + frequencySlope \cdot frequency. \quad (38.3)$$

The slope fit of frequencySlope = 0.01 means that the expected probability of occurrence of a word on day 101 is equal to the observed proportion of days that word occurred in the past 100 days. This indicates a slowly changing information environment, with a low rate of new words in *The New York*

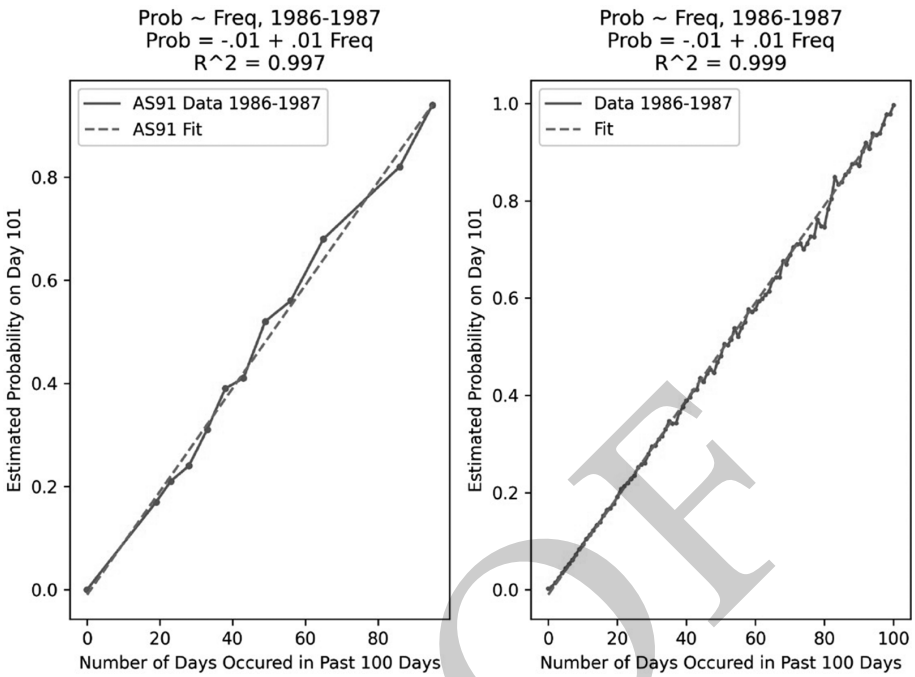

Figure 38.2 *The relationship between frequency and need probability based on the ecological statistics of words in* The New York Times *between 1986 and 1987*
Notes: (a) AS91 refers to the analyses presented in Anderson and Schooler (1991). We have replotted the values here based on the values published in that paper. (b) Comparable plots based on our methodology where the data are gathered from The New York Times*'s API. In both cases, the fits are comparable and the probability a word will be needed increases with its frequency over the last 100 days.*

Times's headlines over the scale of 100 days. This may seem trivial, but is in contrast to Anderson and Schooler (1991)'s findings of faster changing information environments in parental speech (frequencySlope = 0.0076) and email (frequencySlope = 0.009).

Figure 38.3 provides a more direct analysis of the practice effect as a power law by investigating the relationship between the log odds of the probability the word will be needed $\log\left(\frac{p}{1-p}\right)$ given its past frequency. It enables a more fine-grained examination of the tail behaviour, as most words occur at low frequencies. Without the log odds conversion, the differences in the distribution of low frequency words are more challenging to see. Qualitatively, the two plots look similar. Our analysis has more precision, which is expected given the computational limitations faced by Anderson and Schooler (1991). Quantitatively, we fit a linear regression to the first 50 values (as the tail skews the analysis drastically otherwise – a more sophisticated approach could be a bootstrapped linear regression):

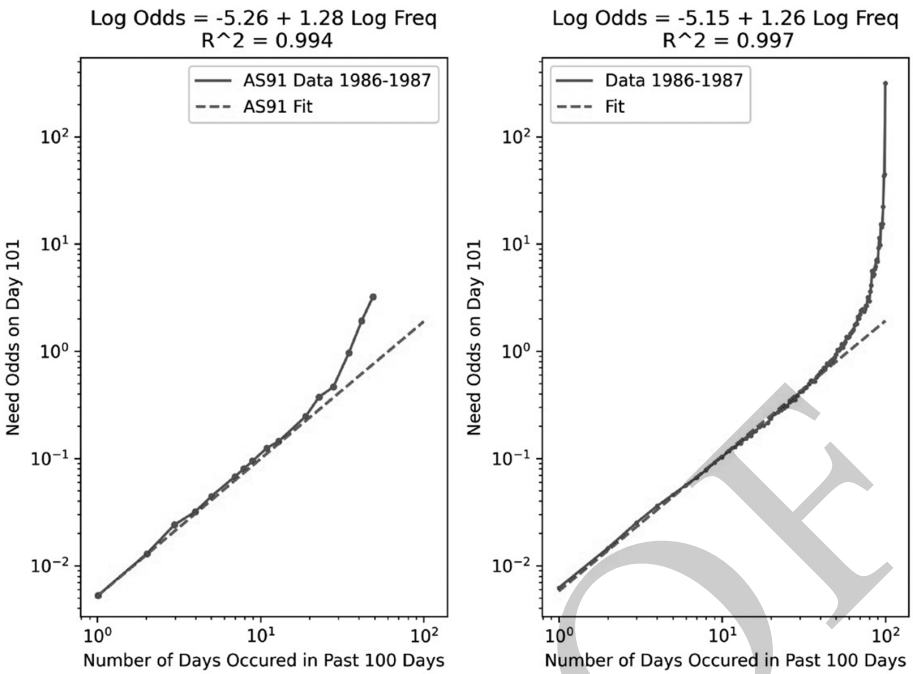

Figure 38.3 *The relationship between frequency of daily occurrences and need odds based on headlines in* The New York Times *between 1986 and 1987*
Notes: (Left) The analysis presented in Anderson and Schooler (1991). (Right) Our reproduction of the same analysis based on our methodology where the data are gathered from The New York Times*'s API. In both cases, the fits are comparable. The axes are log transformed and the data appears to follow a power law for word frequencies up to around 50 appearances in the last 100 days, with fits shown for that interval as dashed lines.*

$$log(needOdds) = frequencyLogIntercept + frequencyLogSlope \cdot log(frequency) \tag{38.4}$$

As before, the numerical fits are very similar and, as before, frequencyLogSlope $= 1.27 > 0$ indicates that the more frequent a word, the larger odds that it will be needed.

Next, we examined the recency effect in the statistics for years 1986–1987. To do so, we first visualised the relationship between the number of days since a word last occurred to its probability (relative to day '101' in a sliding window of 100 days). As shown in Figure 38.4, the probability a word will be used given the length of time since its last occurrence decays rapidly. In this case, there were differences between our analysis and Anderson and Schooler (1991). Our analysis has periodic 'spikes'. When examining our data, we found that these spikes occur at multiples of 7 days and were partially due to recurring periodic sections of *The New York Times*. For example, 'weekender', 'bookshelf' and

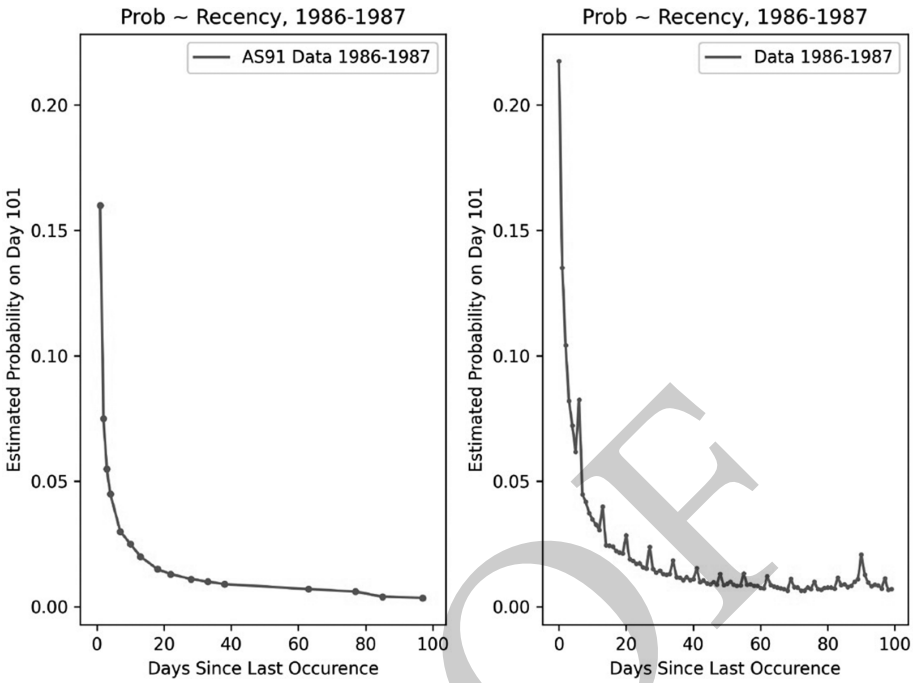

Figure 38.4 *The relationship between how recent the last time a word occurred and the probability it will be needed based on the statistics of words in* The New York Times *between 1986 and 1987.*
Notes: (Left) AS91 refers to the analyses presented in Anderson and Schooler (1991). We have replotted the values here based on the values published in that paper. (Right) Comparable plots based on our methodology where the data are gathered from The New York Times*'s API. In both cases, the fits are comparable and the probability a word will be needed increases with its frequency over the last 100 days.*

'verbatim' all relate to weekly sections that show up with strong 7 day signals in our analysis. Other words that occur weekly in the data set are 'week's', 'wednesday' and 'thursday'.

Figure 38.5 visualises the relationship between the odds of the probability that a word will be needed and the recency of the last occurrence on a log–log scale. Following our prior analyses, we fit a linear regression to the log transformed data:

$$log(recurringOdds) = recencyLogIntercept + recencyLogSlope \cdot log(recency). \tag{38.5}$$

In this case, there are some quantitative differences between prior fits and our own.

However, they are minor and do not change the interpretation. In both cases, recencyLogSlope ≈ -0.75. This indicates that the longer it has been since a word was last used, the less likely it is that it will be used now.

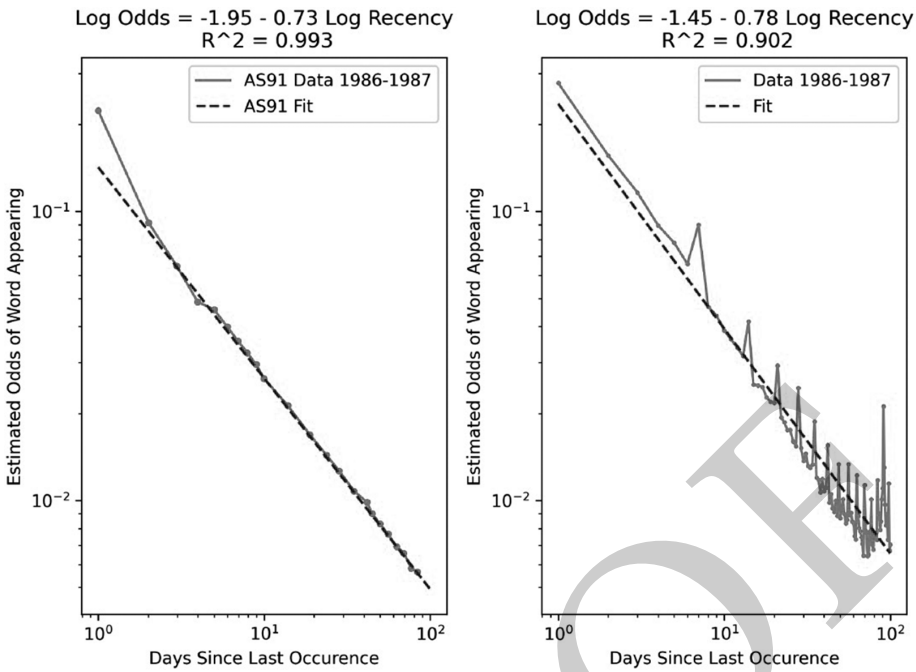

Figure 38.5 *The relationship between the recency of a word and the probability it will be needed plotted in log–log space*
Notes: (a) Results from Anderson and Schooler (1991). (b) Our results. Although there are some differences between the two plots, both are approximately linear in log–log space and have similar fits.

In this section, we replicated some results from Anderson and Schooler (1991) for the years 1986–1987 using a modern data set and methodology. There were a few differences, but none that invalidate either approach, and the converging results should reassure the reader as to their veracity. This highlights the importance of visualising and validating statistics of interest and replication of prior results. Differences during replication are not in themselves a cause for alarm. It is inevitable that there will be some differences between analyses, especially those done decades apart with different sources for the data sets. What is essential is that the interpretation of how the analyses support theories of interest is the same, the qualitative and quantitative differences between the two analyses is easily accessible to a reader and that the differences are interpretable.

38.6 Trends over the Last Century

We are interested in how the environment may have changed over time. To investigate this, we performed analyses of frequency and recency effects in *The New York Times* for every year from 1919 to 2019, following the same

methodology as in Section 38.5. For each year, we fit models to the data and investigated whether the parameters of those models changed over time or not. Different years have different numbers of headlines, which we found could have an effect on the analyses. We controlled for this by randomly selecting 70,000 headlines for each year (which is less than the lowest headline count for any year). We found that the year 1978 was anomalous, with parameters far outside the range of the other years. After investigating, it seems that there was a newspaper strike in New York in 1978 that severely interrupted publishing that year (Traub, 2020). For this reason, we removed this year from the analysis.

In both the frequency and recency analyses we followed Anderson and Schooler (1991) by considering a window size of 100 days. We also checked other window sizes and found similar results as reported here. This is an important analysis as it ensures our arbitrary decision of a window size of 100 days (as opposed to 102) is not determining our results. A first step in trend analysis is plotting the data and inspecting it visually. Linear regression is a useful step to quantify the relationship between time and changes in the parameter of interest. We report the fits from linear regression trends but opted for a more flexible approach to test for significant trends. To do this, we analysed the time series using the Mann-Kendall trend test (Hussain & Mahmud, 2019), which tests for the presence of a monotonic trend against the null hypothesis of no trend. The Mann-Kendall test is non-parametric so that it does not assume a specific form of the data beyond the points being sampled independently. A p-value below 0.05 means that we can reject the null hypothesis of no trend at a 5 per cent significance level.

We began by looking at the relationship between the number of days (frequency) that a word occurred in the last 100 days and the estimated probability of word occurrence on the 101st day (Figure 38.6). There seems to be an overall downward trend in this relationship (Mann-Kendall $p = 0.001$), as parameterised by the constant of proportionality between frequency and probability (frequencySlope $= 0.0114 - 8 \times 10{-}7$ years). In one sense this parameter is a measure of change in language use. For example, text generated from a static bag of words model would result in a constant parameter value of 0.01. A negative trend indicates that usefulness of frequency as a cue for prediction is decreasing.

To illustrate the differences in the fitted frequency–probability model shown in Figure 38.6, we show two years side by side in Figure 38.7. We selected 1920, the first year in our data set with a fitted parameter of 0.0099, and 2012, a recent year that has the lowest fit of the data set, with a parameter of 0.0095. As discussed, a parameter of 0.01 indicates maximum predictability and minimal change in word use. This difference can be clearly seen in Figure 38.7, with 2012 having much more noise as well as a lower slope overall.

A similar pattern was found when investigating the frequency effect in terms of the relationship between the frequency of daily occurrence in the past 100 days and the estimated need odds on Day 101 (Figure 38.8) (frequencyLogSlope $= 1.6144 - 0.0002$ years). As in Anderson and Schooler

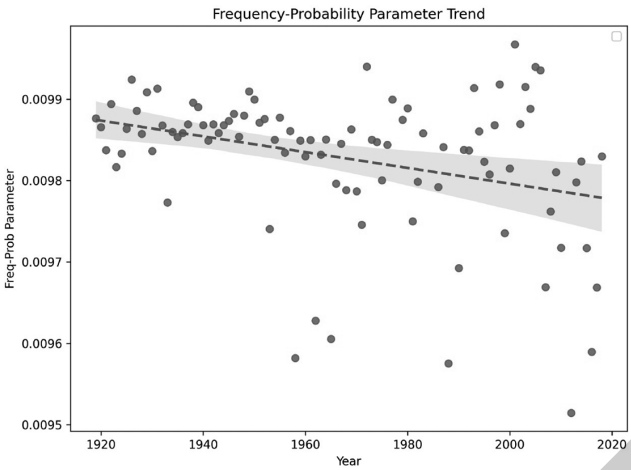

Figure 38.6 *The trend in fitted model parameters between frequency of daily occurrences of* The New York Times *headline words in a window of 100 days and estimated probability of word occurrence on day 101*
Notes: The dashed line shows a linear regression fit for the trend, with the shaded region being a bootstrapped 95% confidence interval. Data for 1978 was removed for the plot and analysis, as anomalous data.

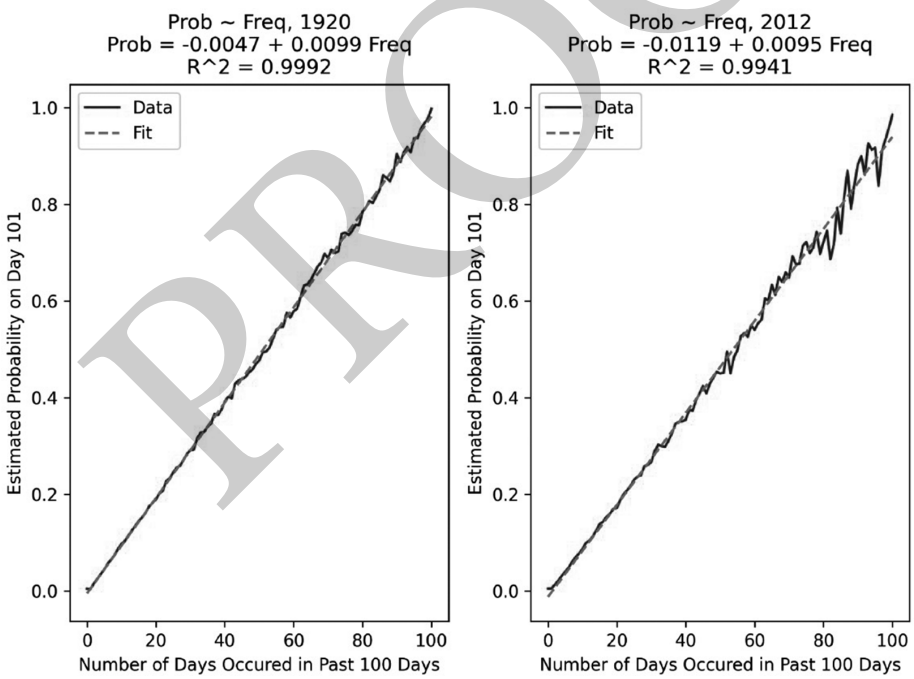

Figure 38.7 *The fitted model between the number of days a word occurred in* The New York Times *headline in a window of 100 days (frequency) and the estimated probability of that word occurrence on day 101*
Notes: In 1920 (left), the word occurrence on day 101 was much more predictable than in 2012 (right), with 1920 having a closer fit and a gradient closer to 0.01.

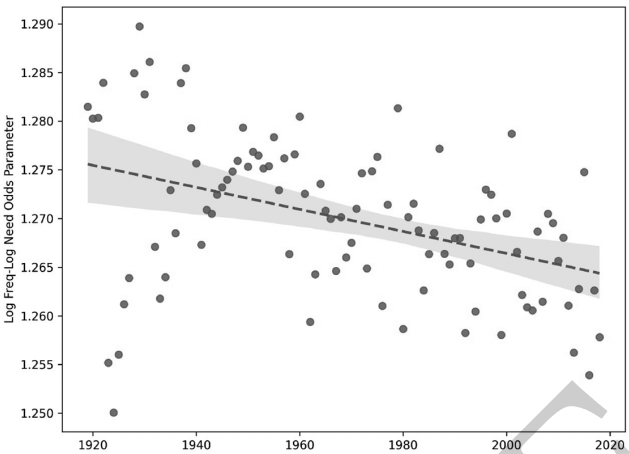

Figure 38.8 *The trend in fitted model parameters between the log of the number of days occurred of* The New York Times *headline words in a window of 100 days and estimated log need odds of word occurrence on day 101*
Notes: The dashed line is a linear regression fit, with the shaded region being a bootstrapped 95% confidence interval. The year 1978 was excluded as it is anomalous.

(1991), the model was fit to the log transformed data considering frequencies up to 50 out of 100 days. This downward trend (Mann-Kendall $p = 0.001$) in the parameter is consistent with the downward trend in the relationship between frequency and probability, as odds is an increasing function of probability.

In total these results suggest that, over the last century, the effect size of frequency (defined as the number of daily occurrences in the last 100 days) has decreased. This indicates that frequency has become a less useful cue for event prediction than 100 years ago. Thus, the environmental demands for human memory have changed such that human memory, if adapting optimally to the environment, should have decreased its reliance on frequency of past occurrences over the last century. For example, when conversing with friends about current events, online language production relies on human memory to predict events that are likely to be discussed.

The trend in recency effect was analysed following a similar method. For each year, a model was fit to the log transformed days since occurrence to the estimated probability of occurrence, as in Anderson and Schooler (1991). Figure 38.9 shows the trend from 1919 to 2019.

There is a clear upward trend (Mann-Kendall $p = 0.001$), with the recency parameter increasing over the last century (recencyLogSlope $= -5.5042 +$ 0.0023 years), starting from around -1.05 in 1919 to around -0.85 in 2019. Changes over the last century are on a much greater scale than the year-to-year noise over that period. This suggests that there have been changes in the information environment over time that may present challenges for human memory. As a larger value indicates a stronger recency effect, this suggests that the

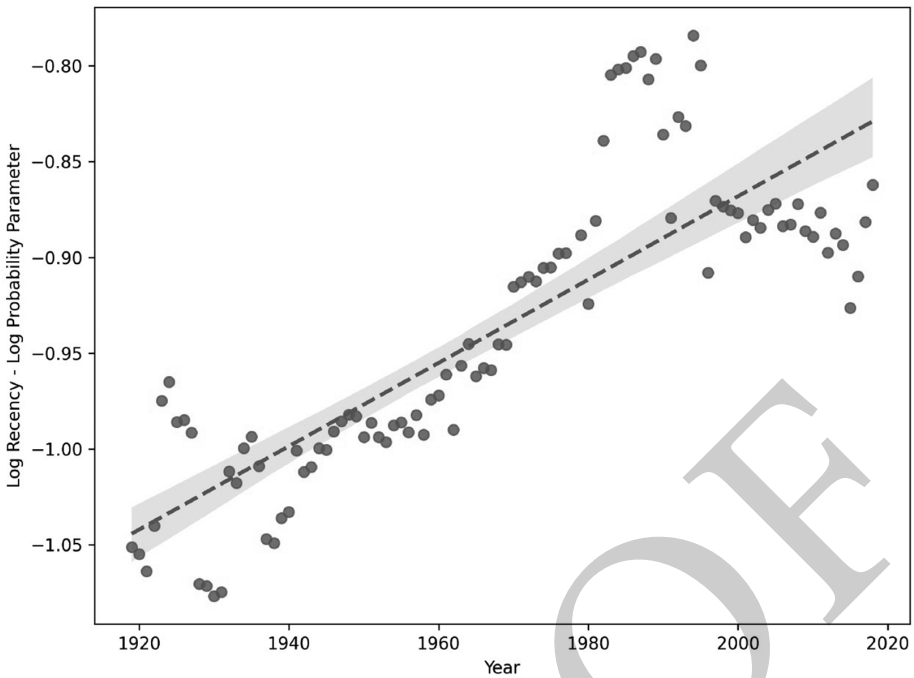

Figure 38.9 *The trend in fitted model parameters between log recency of* The New York Times *headline words estimated log probability of word occurrence on the following day*
Notes: The dashed line is a linear regression fit, with the shaded region being a bootstrapped 95% confidence interval. The year 1978 was excluded as anomalous.

amount of time between event occurrences have been decreasing (on average). In other words, the events of our world are occurring in shorter intervals – events are 'burstier'. We leave it to future work to compare journal articles reporting recency effect sizes from over the last century to validate these predictions.

However, it is also possible that these changes are reflecting people's perception of newsworthy events and not events more broadly. In summary, the analyses suggest that (1) event frequency is becoming a less reliable predictive cue and (2) time between event occurrences is decreasing in size. Events are more likely to occur in more rapid and dense bursts but are less likely to re-occur in the far future. This provides empirical support for the intuition that the world's pace has increased while it increasingly forgets the past.

38.7 Conclusion

Human behaviour can be considered as an adaptive solution to the problems posed by the environment. It follows that analysing the structure of

the environment can improve our understanding of human behaviour. When it comes to cognition, the environment can be information itself. There is a huge wealth of data pertaining to the information that is produced and consumed in human culture.

As such, data science is a key tool in collecting, processing and analysing these large data sets. This can generate insights into human behaviour that can enhance our understanding, test models and generate hypotheses.

Here, we took seriously the idea that human memory is optimised for the structure of events in the world. We replicated earlier findings from Anderson and Schooler (1991) with a larger data set of headlines from *The New York Times*. The dynamics of human memory are similar to what would be expected from an optimal rational agent in the same environment. To be efficient, human memory has to differentially remember or forget things depending on how relevant those things will be in the future.

We found trends in both the frequency and recency effects of events in *The New York Times* over the last 100 years. The frequency trend is statistically significant, although relatively small compared to the noisy oscillations from year to year. This would be consistent with a more dynamic event structure over time, such that events within a 100 day period become less predictive of events on day 101. The recency trend is stronger, with a clearer upward trend in the recency parameter (becoming less negative). Both the frequency and recency trends suggest a more dynamic event structure over time, which is consistent with other work that suggests that the rate of change of public discussion is increasing in the modern world (Lorenz-Spreen et al., 2019). Our work joins a growing literature analysing large text corpora to gain insights in historical changes in social sciences (Li et al., 2019; Michel et al., 2011).

Analysing large amounts of text is known in the digital humanities as distant reading, as contrasted to close reading where one carefully considers and reflects on a piece of writing (Jänicke et al., 2015). In order to detect trends in human language, there simply is not enough time to closely read everything produced. Human language is rich and complex and much of the information contained within is lost when carrying out computational analysis of these types.

During both the frequency and recency analysis we reduced an entire year's headlines to one number, and then reduced an entire century of headlines to a trend line. This extreme compression loses much information. However, we had good reasons behind the analysis at each step, so that the information that was captured tells us something fundamental about the structure of human events each year, and how that changes. In this way distant reading does not replace close reading as a research tool but complements it and offers new perspectives on the human information environment.

We assumed here that the headlines from *The New York Times* represent some independent measure of events in the world. Although illuminating, there are still some fundamental limitations. As discussed in this chapter, assuming each word is an event, which is too strong of an assumption. Better methods for deriving events from headlines should be developed and validated to assess the

robustness of our findings. Additionally, the editorial board of *The New York Times* has likely changed how it decides what to publish and how often to publish it. It is possible that our results reflect changes due to the editorial board rather than the event dynamics in the last century.

Although possible, it would not explain why the particular changes in event dynamics happened at the times they did. If it were due to changes in how the editorial board made decisions, event statistic changes should be likely to happen at each major change in the editorial board. However, trends in the event statistics were more consistent than this would predict.

Of course the headlines themselves are generated partly by human cognition and biases (as are child directed speech and emails, the other measures used by Anderson and Schooler (1991)). Thus, the events are likely changing due to human actions (e.g., the rapid changes in technology and globalisation). The environment is not fixed but is the result of complex interactions between the world and many individuals, each with their own memory dynamics. This can be thought of as a form of niche construction, whereby an organism changes their environment in a way that also changes the adaptive landscape for that organism (or other organisms), which is especially potent in human cultural development (see Kendal et al., 2011). What does this mean for our analysis?

Fundamentally, the cultural environment still presents a problem for human memory to solve, regardless of the specific causal chain that created the environment. As such, the assumptions underpinning rational analysis hold.

The modern world has, in recent history, being changing more rapidly than human evolutionary timescales. We find ourselves in an environment which is quite different from the environment of evolutionary adaptedness (Irons, 1998). Humans do have a wonderful ability to adapt to a range of environments, although it is perhaps unclear the degree to which certain behaviours can bend or how that affects our well-being. If the modern world is changing rapidly, the information environment is leading the way. As Simon (1969) stated, the modern abundance of information creates a poverty of human attention, which is competed for more and more vigorously in the attention economy. That in itself drives further changes in the information environment and more competition (Hills & Adelman, 2015). In some sense, this competition for attention may drive information to be a better fit for human cognition, to be more easily consumable (although perhaps not more beneficial).

References

Anderson, J. R. (1990). *The adaptive character of thought*. Lawrence Erlbaum Associates.

Anderson, J. R. & Milson, R. (1989). Human memory: An adaptive perspective. *Psychological Review*, 96(4), 703.

Anderson, J. R. & Schooler, L. J. (1991). Reflections of the environment in memory. *Psychological Science*, 2(6), 396–408.

Burrell, Q. L. (1985). A note on aging on a library circulation model. *Journal of Documentation*, 41, 100–115.

Davies, M. (2010). *The Corpus of Historical American English (COHA)*. Available at https://english-corpora.org/coha/ (last accessed 22 October 2025).

Ebbinghaus, H. (1885). *Über das gedächtnis: untersuchungen zur experimentellen psychologie*. Duncker & Humblot.

Fiser, J. & Aslin, R. N. (2002). Statistical learning of new visual feature combinations by infants. *Proceedings of the National Academy of Sciences*, 99(24), 15822–15826.

Geisler, W. S. (2008). Visual perception and the statistical properties of natural scenes. *Annual Review of Psychology*, 59, 167–192.

Gould, S. J. & Lewontin, R. C. (1979). The spandrels of San Marco and the panglossian paradigm: A critique of the adaptationist programme. *Proceedings of the royal society of London. Series B. Biological Sciences*, 205(1161), 581–598.

Harlow, L. L. & Oswald, F. L. (2016). Big data in psychology: Introduction to the special issue. *Psychological Methods*, 21(4), 447.

Hills, T. T. (2019). The dark side of information proliferation. *Perspectives on Psychological Science*, 14(3), 323–330.

Hills, T. T. & Adelman, J. S. (2015). Recent evolution of learnability in american english from 1800 to 2000. *Cognition*, 143, 87–92.

Hopman, E., Thompson, B., Austerweil, J. & Lupyan, G. (2018). Predictors of l2 word learning accuracy: A big data investigation, in *40th Annual Conference of the Cognitive Science Society (CogSci 2018)*. Cognitive Science Society, 513–518.

Hussain, M. M. & Mahmud, I. (2019). Pymannkendall: A python package for non parametric Mann Kendall family of trend tests. *Journal of Open Source Software*, 4(39), 1556.

Irons, W. (1998). Adaptively relevant environments versus the environment of evolutionary adaptedness. *Evolutionary Anthropology: Issues, News, and Reviews: Issues, News, and Reviews*, 6(6), 194–204.

Jänicke, S., Franzini, G., Cheema, M. F. & Scheuermann, G. (2015). On close and distant reading in digital humanities: A survey and future challenges. *EuroVis (STARs)*, 2015, 83–103.

Jones, M. & Love, B. C. (2011). Bayesian fundamentalism or enlightenment? On the explanatory status and theoretical contributions of Bayesian models of cognition. *Behavioral and Brain Sciences*, 34, 169–231.

Kendal, J., Tehrani, J. J. & Odling-Smee, J. (2011). Human niche construction in interdisciplinary focus. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 366(1566), 785–792.

Kim, A. S. N., Wong-Kee-You, A. M. B., Wiseheart, M. & Rosenbaum, R. S. (2019). The spacing effect stands up to big data. *Behavior Research Methods*, 51, 1485–1497.

Kingman, J. F. C. (1992). *Poisson processes*. Clarendon Press.

Kuzemko, C., Blondeel, M., Bradshaw, M., Bridge, G., Faigen, E. & Fletcher, L. (2025). Rethinking energy geopolitics: Towards a geopolitical economy of global energy transformation. *Geopolitics*, 30(2), 531–565.

Li, Y., Engelthaler, T., Siew, C. S. & Hills, T. T. (2019). The macroscope: A tool for examining the historical structure of language. *Behavior Research Methods*, 51(4), 1864–1877.

Lorenz-Spreen, P., Mønsted, B. M., Hövel, P. & Lehmann, S. (2019). Accelerating dynamics of collective attention. *Nature Communications*, 10(1), 1–9.

MacWhinney, B. & Snow, C. (1990). The child language data exchange system: An update. *Journal of Child Language*, 17, 457–472.

Michel, J. B., Shen, Y. K., Aiden, A. P., et al. (2011). Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014), 176–182.

Mitzenmacher, M. (2004). A brief history of generative models for power law and lognrmal distributions. *Internet Mathematics*, 1(2), 226–251.

Mozer, M. C., Pashler, H., Cepeda, N., Lindsey, R. V. & Vul, E. (2009). Predicting the optimal spacing of study: A multiscale context model of memory, in *Advances in neural information processing systems*, Vol. 22, edited by Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams & A. Culotta. Curran Associates, 1321–1329.

Newell, A. & Rosenbloom, P. (1981). Mechanisms of skill acquisition and the law of practice, in *Cognitive skills and their acquisition*, edited by J. R. Anderson. Lawrence Erlbaum Associates, 1–55.

Paxton, A. & Griffiths, T. L. (2017). Finding the traces of behavioral and cognitive processes in big data and naturally occurring datasets. *Behavior Research Methods*, 49, 1630–1638.

Pilgrim, C., Guo, W. & Hills, T. T. (2021). Information foraging in the attention economy drives the rising entropy of English. *arXiv*.

Romberg, A. & Saffran, J. (2010). Statistical learning and language acquisition. *Wiley Interdisciplinary Reviews: Cognitive Science*, 1(6), 906–914.

Saffran, J., Aslin, R. N. & Newport, E. (1996). Statistical learning by 8-month-old-infants. *Science*, 274(5294), 1926–1928.

Schooler, L. J. & Anderson, J. R. (1997). The role of process in the rational analysis of memory. *Cognitive Psychology*, 32, 219–250.

Schütze, H., Manning, C. D. & Raghavan, P. (2008). *Introduction to information retrieval*, vol 39. Cambridge University Press.

Simon, H. A. (1955). On a class of skew distribution functions. *Biometrika*, 52, 425–440.

(1969). *Designing organizations for an information-rich world*. Brookings Institute Lecture.

Traub, A. (1 April 2020). When all the zingers were fit to print. *The New York Times*, A25.

Wickelgren, W. A. (1976). Memory storage dynamics, in *Handbook of learning and cognitive processes*, edited by W. K. Estes. Lawrence Erlbaum Associates, 321–361.