

Media Memory

Database Design
Spring 2007

UCLA departments of Statistics, Design and Media Arts, and Information
Science

Group Participants:

Aaron Siegel
Katie Shilton
Simran Khalsa
Yunsil Heo

Thoughtful Foundations

Research conducted by Katie Shilton

Media Memory questions how media lenses shape social memory and forgetting. Does frequency of media attention to a topic play a role in fixing an event in social memory? And do these memories degrade over time as they are replaced by new subjects of attention?

News media has a complicated role in the production of social memory. Journalists engage in a variety of activities which reference and interpret historical events, writing and re-writing topics into a larger social consciousness. For example, reporters write commemorations of historical events, invoke historical analogies to describe present-day events, and situate current events within interpreted historical contexts (Edy, 1999). Barry Schwartz suggests that "commemorative symbolism" in outlets like media news can help to alter how historical events are remembered (Schwartz, 1997).

And media is involved not only in memory, but also in forgetting. The research of Thijs Pollmann (1998) suggests measurable "forgetting" curves within the news media. Pollmann tallied references to calendar years in international newspaper texts and found "empirical evidence that corroborates the intuition that the further the past recedes, the less often we make an appeal to it..." (p. 320).

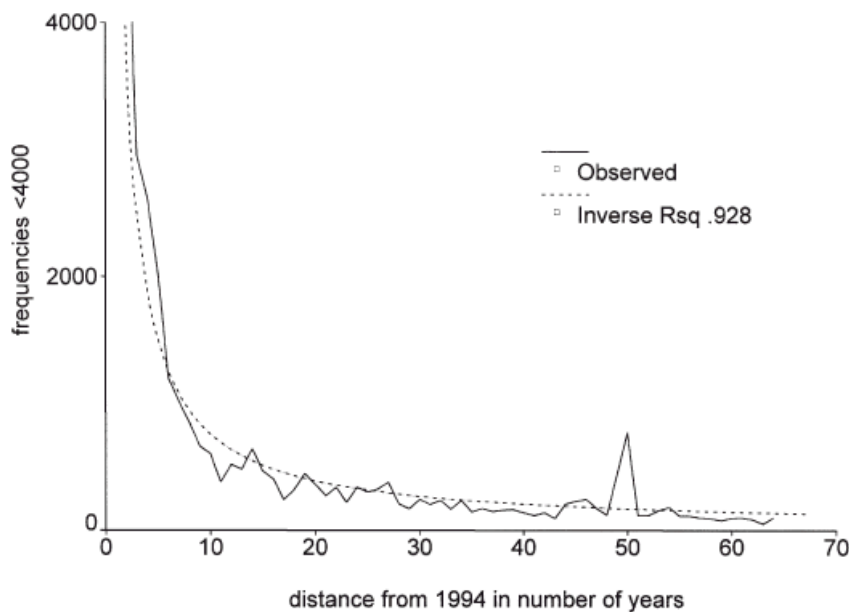


Figure 2. *International Herald Tribune* 1994: Time series 1993–1930.

(Pollmann, 1998, p. 323)

The effect of media on social memory ultimately depends on the way that we interpret the meaning of media communication. Some researchers propose that the media operates in a "transmission" model, in which producers choose content to broadcast to the masses (McQuail, 2005). In a transmission model, Media Memory becomes a direct representation of what the media (and therefore, society) remembers, showing the most recent and frequent topics of media attention writ large.

Media Memory, however, suggests that interpreting media memory is more complicated than a transmission model. As Media Memory shows, topics that grow and fade over time lose meaning when removed from their context. Our ability to decode the story behind the topic disappears when a headline is represented as merely the frequencies of its component parts. Media Memory evokes media memory within a "reception" model (McQuail, 2005). Media producers may select topics and headlines, but consumers actively interpret these selections. Making sense of news becomes an individual, contextual process. Individual sense-making does not reject a role for mass media in collective memory. It merely complicates the equation of actors and processes that shape what is remembered and forgotten in media headlines.

References

- Edy, J. A. (1999). Journalistic uses of collective memory. *Journal of Communication*, 49 (2), 71-85.
- McQuail, D. (2005). *McQuail's Mass Communication Theory*. London, Thousand Oaks and New Delhi: Sage Publications.
- Pollmann, T. (1998). On forgetting the historical past. *Memory & Cognition*, 26 (2), 320-329.
- Schwartz, B. (1997). Collective memory and history: How Abraham Lincoln became a symbol of racial equality. *The Sociological Quarterly*, 38(3), 469-496.

User Experience

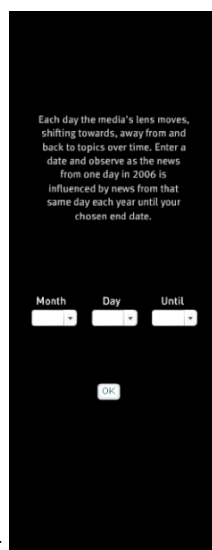
A product of cumulative group discussion, decision-making and creative application.

A dark screen offers three white rectangles to solicit a day, a month and a year from the user. There is a moment of anticipation, complete darkness except for the white letters of the date at the very bottom of the screen, then words begin to fill in the space. First the words from the single chosen day of the most recent archived calendar year, 2006. The most frequent terms are the boldest, the least frequent smaller and faded. Then terms from the year before, in a different color fill in more of the dark space. Then terms from the year before that, in another different color, fill in. The cycle continues back till it hits the user-chosen year limit.

The cycle stops, the darkness is completely filled with color and words, a different color for each year, a different intensity of color for the different relative frequencies of the words.

Some words repeat. Some words stand alone.

The final image is a conglomeration of colorful words that fill in a slender column of darkness. It is a pillar of media memory. Intense words, repeated words, stand out. The rest fade back, as they do in our own memories.



Begin at

enter 07/28/1997, and end up with



Display Module

Coding and display by Yunsil Using Flash CS3

Written in Flash, the display module uses the standard calendar function to solicit a date from the user, which it passes on to the backend using XML.

The date range has a lower limit of 1988, because beyond then the google news archive fails to return consistent data for the current query structure.

When the results are returned as an XML document (for further information see the backend, and the example return XML document), an Action Script extracts the words and frequencies from the XML, then uses a modified form of Jared Tarbell's box-fitting algorithm, whose open-source code is available from www.levitated.net, to create the display. The backend results are returned one year at a time, with the most recent archived year (currently 2006) first. The results are displayed cumulatively, bringing in the memory of years past.

Each year's words appear in a different color. The first color is a random selection, the rest of the colors maintain a fixed ratio of HSV (hue, saturation, value) based upon that first choice so that the cumulative image's color balance is harmonious. For each color, each year, the most frequently occurring words appear at full color saturation. The saturation levels drop with the relative frequency, giving a faded look to less used words.

See the included cd for the flash files, screen shots of completed displays for several selected dates, and a movie (.avi) of the Media Memory module in action.

Backend Module

Coding and scraping by Aaron Siegel

Run in a hosting environment compatible with Python 2.3

Using the python scripting language and common gateway interface (cgi) protocol, the backend coding takes the user-provided date, June 12, 1994, for example, and queries the google news archive for the top 20 articles on that date for 2006, 2005, 2004, 2003.... 1994. The queries are broken down to ask for results from each year independently so that the data will be returned in chunks, distributing the bandwidth and processing load and supporting the function of the display module.

The google news archive returns both an individual title and abstract for each of the top twenty articles, as well as a weighting frequency for each article, in the form of, '320 more like this.' These values are all utilized by the script.

The script filters out noise - such as common stop words - and letter casing. Numerical language modeling is applied to the resulting pool of lower case words. They are sorted into a list of words and their frequencies. All words in and individual title and abstract are weighted equally. The abstract is guaranteed to contain the 'about-ness' of an article, but the headline contains the most noticed words to someone skimming the news, so both aspects contribute heavily to the media impact on memory. The individual word weight is then informed by the number of 'articles like this' that the google archive returned.

The script assembles the words and frequencies into an XML document using a schema developed for this application. An XML document is returned to the display module for each year in the range.

Backend Python Script

```
#!/usr/bin/python
# -*- coding: latin-1 -*-

import httplib, re, cgi
from datetime import datetime
from BeautifulSoup import BeautifulSoup

class Engine:
    def __init__(self):
        self.words = {}
        self.articlelist = []
        self.junkchars = [".", ",", ";", ":", " " | "[", "]", "_", "-", "(", ")", "\\", "&"]
        self.noisewords = file("noise.txt", "r").read().split(" ")
        print "Content-Type: text/xml\n"
        #print "Content-Type: text/plain\n"
        form = cgi.parse()          # seperate query string into dictionary
        day = int(form["day"][0])   # day
        month = int(form["month"][0]) # month
        year = int(form["year"][0]) # year
        now = datetime.now()       # get current date
        if now.year == year:
            print "not now!"
            #self.searchNew()
        else:
            self.searchOld(year, month, day)
        wordlist = self.sortWords()
        self.formatXML(wordlist, year, month, day)

    def analyze(self, article):
        wordlist = article["headline"].split()
        wordlist.extend(article["abstract"].rstrip("...").split())
        for word in wordlist:
            #cleanword = word.lower()
            cleanword = self.clean(word.lower())
            if cleanword not in self.noisewords and len(cleanword) > 0:
                if self.words.has_key(cleanword):
                    self.words[cleanword] += 1
                else:
                    self.words[cleanword] = 1
```

```

def clean(self, word):
    word = word.replace("&quot;", "")
    for char in self.junkchars:
        word = word.strip(char)
    return word

def download(self, page):
    host = "news.google.com"
    h = httplib.HTTP(host)
    h.putrequest("GET", page)          # define page to request
    h.putheader("Host", host)         # define host
    h.putheader("User-agent", "media-memory") # define user agent
    h.endheaders()                   # apply
    returncode, returnmsg, headers = h.getreply() # get data
    f = h.getfile()                   # convert to file format
    h.close()                         # close connection
    data = f.read().decode("latin-1") # read from file format
    return data

def formatXML(self, words, year, month, day):
    print "<data year=\ ""+str(year)+"\ " month=\ ""+str(month)+"\ "
day=\ ""+str(day)+"\ ">"
    print "<headlines>"
    for article in self.articlelist:
        print "<h>"
        print "<title>"
        print article["headline"]
        print "</title>"
        print "<abstract>"
        print article["abstract"]
        print "</abstract>"
        print "<num>"
        print str(article["num"])
        print "</num>"
        print "</h>"
    print "</headlines>"
    print "<words>"
    for word in words:
        print "<w word=\ ""+word[0)+"\ " num=\ ""+str(word[1])+"\ "/>"
    print "</words>"
    print "</data>"

def parseRSS(self, rss):
    soup = BeautifulSoup()

```



```

soup.feed(rss)
news = []
for item in soup("item"):
    thing = {}
    tempstuff = item.title.contents[0].encode("latin-1",
"xmlcharrefreplace").split(" - ")
    thing["source"] = tempstuff.pop()
    thing["title"] = " - ".join(tempstuff)
    thing["date"] = item.pubdate.contents[0].encode("latin-1")
    thing["link"] =
item.contents[2].split("url=",1)[1].split("&",1)[0].encode("latin-1")
    if item.description.contents[0].find("imgurl=") > 0:
        step1 = item.description.contents[0].split("imgurl=",1)[1]
        url,step2 = step1.split(" width=",1)
        thing["img"] = "http://"+url.encode("latin-1")
    else:
        thing["img"] = None
        step2 = item.description.contents[0];
    if step2.find("&gt;all ") > 0:
        thing["num"] = step2.split("&gt;all ",1)[1].split(" ",1)[0].encode("latin-1",
"xmlcharrefreplace")
    else:
        thing["num"] = "1"
    print thing["num"] + "\t" + thing["title"]
    news.append(thing)
return news

def requestPage(self, engine, query):
    html = self.download(engine+query)
    content = html.split("of about <b>",1)[1]
    resultnum, content = content.split("</b>",1)
    soup = BeautifulSoup()
    soup.feed(content)
    #print soup.prettify()
    return resultnum, soup
# make soup object
# feed soup the article content

def searchNew(self):
    resultnum, soup = self.requestPage("/news?num=20&q=", ".")
    for result in soup.findAll("td", attrs={"class":"j"}):
        url = result.find("a")["href"]
        title = ""
        for piece in result.find("a").contents:
            title += str(piece).replace("<b>", "").replace("</b>", "")
        print title

```

```

print "new articles > " + resultnum

def searchOld(self, year, month, day):
    date = str(month)+"%2F"+str(day)+"%2F"+str(year)
    string =
"/archivesearch?num=20&as_ldate="+date+"&as_hdate="+date+"&q="
    resultnum, soup = self.requestPage(string, ".")
    #print "old articles: " + resultnum + "\n"
    for result in soup.findAll("table", attrs={"style":re.compile("^margin-left*")}):
        #print str(result.tr.td.prettify())
        article = {}
        article["headline"] = str(result.tr.td.a.contents[0]).replace("&#39;", "")
        article["abstract"] =
str(result.tr.td.table.tr.td.font.contents[0]).replace("&#39;", "")+"..."
        article["num"] = result.find(text=re.compile("^All *[0-9]* related")).split()[1]
        self.articlelist.append(article)
        self.analyze(article)

def sortWords(self):
    wordlist = self.words.items()
    wordlist.sort(self.sortFunc)
    #for word in wordlist:
        #print word
    return wordlist

def sortFunc(self,a,b):
    return cmp(b[1],a[1])

e = Engine()

```

Sample XML output of the Media Memory scrape of the google news archive for the date 9/11/2001

-
<data year="2001" month="9" day="11">

-
<headlines>

-
<h>

-
<title>

Global markets plunged in the wake of the attack at the heart of New...

</title>

-
<abstract>

With the linked Global financial markets plunged into chaos following the horrific terrorist attacks on Washington and New York, nervous investors, ...

</abstract>

<num>

163

</num>

</h>

-
<h>

-
<title>

Attacks Were an Act of War -- But Who's the Enemy?

</title>

-
<abstract>

[FINAL Edition]. When Ramzi Yousef, the mastermind of the 1993 World Trade Center bombing, was captured in Pakistan and flown into New York City in a ...

</abstract>

<num>

243

</num>

</h>

-
<h>

<title>
Attack on the United States
</title>

-

<abstract>

NEW YORK -- A coordinated terrorist Attack on the United States Tuesday destroyed the World Trade Center in NEW YORK City, damaged the Pentagon in ...

</abstract>

<num>

120

</num>

</h>

-

<h>

<title>
New York Looks Destruction In The Eye
</title>

-

<abstract>

New York - At just before 9:00 AM, foot traffic on The streets of Manhattan drew to a standstill, with hardened urban dwellers chatting on their cell phones ...

</abstract>

<num>

182

</num>

</h>

-

<h>

<title>
Why the towers collapsed
</title>

-

<abstract>

By Bill Wyman Sept. 11, 2001 | the World Trade Center's twin towers were the tallest buildings in the World at the time of their opening in 1970. ...

</abstract>

<num>

41

</num>

</h>

-
<h>

-
<title>

Tennis: US Open - Hewitt blows hot after the storm; First Grand Slam...
</title>

-
<abstract>

Tennis: US Open - Hewitt blows hot after the storm; First Grand Slam win allows
Australian to come of age as defeated Sampras takes solace in resurgence of ...
</abstract>

<num>

64

</num>

</h>

-
<h>

<title>

For Bonds, The Catch Is Getting Pitches

</title>

-
<abstract>

[FINAL Edition]. San Francisco's Barry Bonds, a menacing figure in The batter's
box, waved his bat back and forth, glaring out at The Colorado pitcher. ...
</abstract>

<num>

64

</num>

</h>

-
<h>

<title>

Rejoicing in the streets of Jenin

</title>

-
<abstract>

While many Palestinians celebrate the attack on the US, Yasser Arafat denounces
it as "unacceptable" and Israelis mourn. By Flore de Préneuf Sept. ...

</abstract>

<num>

96

</num>

</h>

-

<h>

<title>

The Politics of Panic

</title>

-

<abstract>

The summer has barely ended, but President Bush and his Congressional allies are already frantically rewriting their script for The fall. ...

</abstract>

<num>

98

</num>

</h>

-

<h>

-

<title>

REIGN OF TERROR CONCERTED ATTACKS HIT WORLD TRADE CENTER AND THE...

</title>

-

<abstract>

NEW YORK - THE twin towers OF THE WORLD TRADE CENTER were destroyed this morning after two airplanes crashed into them, in what President Bush described as ...

</abstract>

<num>

341

</num>

</h>

-

<h>

<title>

The question is: Will we hate back?

</title>

-

<abstract>

It was a day of horrors that beggared understanding, of grief that dwarfed language. a day of heroism and evil, of cascading rage, frustration and ...

</abstract>

<num>

167

</num>

</h>

-

<h>

<title>

Dell Computer Is in the Catbird Seat, for Now

</title>

-

<abstract>

By JOHN SCHWARTZ When Hewlett-Packard announced last week that it planned to join forces with Compaq Computer, creating a combined hardware and services ...

</abstract>

<num>

50

</num>

</h>

-

<h>

-

<title>

Terrorism Strikes In The United States in a Massive Attack

</title>

-

<abstract>

THIS IS a RUSH TRANSCRIPT. THIS COPY MAY NOT BE In ITS FINAL FORM AND MAY BE UPDATED. THIS IS a RUSH TRANSCRIPT. THIS COPY MAY NOT BE In ITS FINAL FORM AND ...

</abstract>

<num>

61

</num>

</h>

-

<h>

-
<title>

ON PRO FOOTBALL; Giants Lose Battle of the Corners
</title>

-
<abstract>

By THOMAS GEORGE We have seen, especially in the last decade, what a defense can do when it has supreme cover cornerbacks. Deion Sanders was so good at ...

</abstract>

<num>

63

</num>

</h>

-
<h>

<title>
BUSH: US TO HUNT DOWN THE ATTACKERS
</title>

-
<abstract>

WASHINGTON - A shell-shocked President BUSH yesterday vowed TO "HUNT DOWN" THE terrorists that attacked New York's World Trade Center and THE Pentagon. ...

</abstract>

<num>

109

</num>

</h>

-
<h>

-
<title>

Monitor: Newspapers in southern Africa comment on the Abuja agreement, ...
</title>

-
<abstract>

Monitor: Newspapers in southern Africa comment on the Abuja agreement, which is designed to end the land crisis in Zimbabwe; ALL the NEWS OF the WORLD. ...

</abstract>

<num>

64

</num>

</h>

-

<h>

-

<title>

Israeli Foreign Minister Address The Concerns of Global Safety

</title>

-

<abstract>

THIS IS A RUSH TRANSCRIPT. THIS COPY MAY NOT BE IN ITS FINAL FORM AND MAY BE UPDATED. THIS IS A RUSH TRANSCRIPT. THIS COPY MAY NOT BE IN ITS FINAL FORM AND ...

</abstract>

<num>

58

</num>

</h>

-

<h>

-

<title>

Addicted to the game The Atlanta Journal and The Atlanta Constitution

</title>

-

<abstract>

What took the guy so long? This was the biggest slam dunk in the NBA since His Highness decided to dribble out of retirement the first time. ...

</abstract>

<num>

46

</num>

</h>

-

<h>

-

<title>

ON TENNIS; Venus Williams Has the Sport Firmly on Her Racket

</title>

-

<abstract>

By SELENA ROBERTS It was early in the sister showdown ON Saturday night when Serena Williams unleashed a spring-loaded backhand that defied physics as It ...

</abstract>

<num>

30

</num>

</h>

-

<h>

-

<title>

Terrorist attacks shake nation and world - Sep. 11, 2001

</title>

-

<abstract>

By Staff Writer Chris Isidore NEW YORK (CNNfn) - The worst Terrorist attack in US history killed thousands, destroyed The world Trade Center in NEW YORK,

...

</abstract>

<num>

101

</num>

</h>

</headlines>

-

<words>

<w word="new" num="11"/>

<w word="world" num="10"/>

<w word="york" num="9"/>

<w word="trade" num="7"/>

<w word="attack" num="6"/>

<w word="final" num="6"/>

<w word="may" num="6"/>
<w word="center" num="6"/>
<w word="bush" num="4"/>
<w word="transcript" num="4"/>
<w word="attacks" num="4"/>
<w word="copy" num="4"/>
<w word="terrorist" num="4"/>
<w word="form" num="4"/>
<w word="rush" num="4"/>
<w word="global" num="3"/>
<w word="destroyed" num="3"/>
<w word="united" num="3"/>
<w word="first" num="3"/>
<w word="tennis" num="3"/>
<w word="towers" num="3"/>
<w word="states" num="3"/>
<w word="slam" num="3"/>
<w word="after" num="3"/>
<w word="president" num="3"/>
<w word="abuja" num="2"/>
<w word="updated" num="2"/>
<w word="bonds" num="2"/>
<w word="africa" num="2"/>
<w word="day" num="2"/>
<w word="streets" num="2"/>
<w word="edition" num="2"/>
<w word="computer" num="2"/>
<w word="southern" num="2"/>
<w word="open" num="2"/>
<w word="agreement" num="2"/>
<w word="newspapers" num="2"/>
<w word="monitor" num="2"/>
<w word="sept" num="2"/>
<w word="comment" num="2"/>
<w word="washington" num="2"/>
<w word="twin" num="2"/>
<w word="blows" num="2"/>
<w word="pentagon" num="2"/>
<w word="city" num="2"/>
<w word="storm" num="2"/>
<w word="williams" num="2"/>
<w word="hewitt" num="2"/>
<w word="last" num="2"/>
<w word="atlanta" num="2"/>

<w word="11" num="2"/>
<w word="markets" num="2"/>
<w word="plunged" num="2"/>
<w word="hot" num="2"/>
<w word="grand" num="2"/>
<w word="2001" num="2"/>
<w word="time" num="2"/>
<w word="deion" num="1"/>
<w word="summer" num="1"/>
<w word="thomas" num="1"/>
<w word="unleashed" num="1"/>
<w word="schwartz" num="1"/>
<w word="just" num="1"/>
<w word="shell-shocked" num="1"/>
<w word="center's" num="1"/>
<w word="cornerbacks" num="1"/>
<w word="enemy?" num="1"/>
<w word="shake" num="1"/>
<w word="battle" num="1"/>
<w word="hate" num="1"/>
<w word="captured" num="1"/>
<w word="bombing" num="1"/>
<w word="mourn" num="1"/>
<w word="retirement" num="1"/>
<w word="solace" num="1"/>
<w word="nervous" num="1"/>
<w word="terrorists" num="1"/>
<w word="killed" num="1"/>
<w word="attackers" num="1"/>
<w word="terror" num="1"/>
<w word="coordinated" num="1"/>
<w word="thousands" num="1"/>
<w word="catbird" num="1"/>
<w word="congressional" num="1"/>
<w word="giants" num="1"/>
<w word="good" num="1"/>
<w word="hit" num="1"/>
<w word="decided" num="1"/>
<w word="batter's" num="1"/>
<w word="who's" num="1"/>
<w word="dell" num="1"/>
<w word="nation" num="1"/>
<w word="sampras" num="1"/>
<w word="game" num="1"/>

<w word="francisco's" num="1"/>
<w word="traffic" num="1"/>
<w word="jenin" num="1"/>
<w word="foot" num="1"/>
<w word="celebrate" num="1"/>
<w word="defied" num="1"/>
<w word="constitution" num="1"/>
<w word="bat" num="1"/>
<w word="standstill" num="1"/>
<w word="ended" num="1"/>
<w word="corners" num="1"/>
<w word="rage" num="1"/>
<w word="cascading" num="1"/>
<w word="manhattan" num="1"/>
<w word="frantically" num="1"/>
<w word="denounces" num="1"/>
<w word="lose" num="1"/>
<w word="addicted" num="1"/>
<w word="guy" num="1"/>
<w word="préneuf" num="1"/>
<w word="looks" num="1"/>
<w word="week" num="1"/>
<w word="hunt" num="1"/>
<w word="financial" num="1"/>
<w word="sep" num="1"/>
<w word="staff" num="1"/>
<w word="phones" num="1"/>
<w word="back" num="1"/>
<w word="seat" num="1"/>
<w word="racket" num="1"/>
<w word="nba" num="1"/>
<w word="morning" num="1"/>
<w word="saturday" num="1"/>
<w word="resurgence" num="1"/>
<w word="israeli" num="1"/>
<w word="compaq" num="1"/>
<w word="yousef" num="1"/>
<w word="chaos" num="1"/>
<w word="crisis" num="1"/>
<w word="cell" num="1"/>
<w word="zimbabwe" num="1"/>
<w word="heroism" num="1"/>
<w word="safety" num="1"/>
<w word="decade" num="1"/>

<w word="fall" num="1"/>
<w word="before" num="1"/>
<w word="forth" num="1"/>
<w word="announced" num="1"/>
<w word="dunk" num="1"/>
<w word="football" num="1"/>
<w word="creating" num="1"/>
<w word="opening" num="1"/>
<w word="york's" num="1"/>
<w word="journal" num="1"/>
<w word="designed" num="1"/>
<w word="early" num="1"/>
<w word="drew" num="1"/>
<w word="understanding" num="1"/>
<w word="address" num="1"/>
<w word="menacing" num="1"/>
<w word="described" num="1"/>
<w word="come" num="1"/>
<w word="venus" num="1"/>
<w word="takes" num="1"/>
<w word="box" num="1"/>
<w word="sister" num="1"/>
<w word="allies" num="1"/>
<w word="language" num="1"/>
<w word="getting" num="1"/>
<w word="many" num="1"/>
<w word="barely" num="1"/>
<w word="allows" num="1"/>
<w word="foreign" num="1"/>
<w word="planned" num="1"/>
<w word="act" num="1"/>
<w word="dwellers" num="1"/>
<w word="palestinians" num="1"/>
<w word="9:00" num="1"/>
<w word="pakistan" num="1"/>
<w word="already" num="1"/>
<w word="colorado" num="1"/>
<w word="figure" num="1"/>
<w word="arafat" num="1"/>
<w word="win" num="1"/>
<w word="tuesday" num="1"/>
<w word="down" num="1"/>
<w word="chris" num="1"/>
<w word="news" num="1"/>

<w word="highness" num="1"/>
<w word="george" num="1"/>
<w word="eye" num="1"/>
<w word="horrific" num="1"/>
<w word="script" num="1"/>
<w word="barry" num="1"/>
<w word="question" num="1"/>
<w word="two" num="1"/>
<w word="wyman" num="1"/>
<w word="worst" num="1"/>
<w word="biggest" num="1"/>
<w word="australian" num="1"/>
<w word="yasser" num="1"/>
<w word="tallest" num="1"/>
<w word="war" num="1"/>
<w word="urban" num="1"/>
<w word="showdown" num="1"/>
<w word="frustration" num="1"/>
<w word="chatting" num="1"/>
<w word="attacked" num="1"/>
<w word="collapsed" num="1"/>
<w word="pitcher" num="1"/>
<w word="pitches" num="1"/>
<w word="reign" num="1"/>
<w word="grief" num="1"/>
<w word="wake" num="1"/>
<w word="minister" num="1"/>
<w word="hardened" num="1"/>
<w word="catch" num="1"/>
<w word="dribble" num="1"/>
<w word="especially" num="1"/>
<w word="horrors" num="1"/>
<w word="1993" num="1"/>
<w word="john" num="1"/>
<w word="join" num="1"/>
<w word="isidore" num="1"/>
<w word="flore" num="1"/>
<w word="bill" num="1"/>
<w word="roberts" num="1"/>
<w word="defense" num="1"/>
<w word="backhand" num="1"/>
<w word="hunt" num="1"/>
<w word="following" num="1"/>
<w word="selena" num="1"/>

<w word="history" num="1"/>
<w word="heart" num="1"/>
<w word="supreme" num="1"/>
<w word="beggared" num="1"/>
<w word="concerted" num="1"/>
<w word="san" num="1"/>
<w word="investors" num="1"/>
<w word="sanders" num="1"/>
<w word="pro" num="1"/>
<w word="defeated" num="1"/>
<w word="unacceptable" num="1"/>
<w word="sport" num="1"/>
<w word="seen" num="1"/>
<w word="politics" num="1"/>
<w word="down" num="1"/>
<w word="linked" num="1"/>
<w word="firmly" num="1"/>
<w word="damaged" num="1"/>
<w word="end" num="1"/>
<w word="night" num="1"/>
<w word="rejoicing" num="1"/>
<w word="rewriting" num="1"/>
<w word="writer" num="1"/>
<w word="concerns" num="1"/>
<w word="forces" num="1"/>
<w word="terrorism" num="1"/>
<w word="glaring" num="1"/>
<w word="panic" num="1"/>
<w word="ramzi" num="1"/>
<w word="massive" num="1"/>
<w word="waved" num="1"/>
<w word="buildings" num="1"/>
<w word="de" num="1"/>
<w word="long?" num="1"/>
<w word="serena" num="1"/>
<w word="airplanes" num="1"/>
<w word="dwarfed" num="1"/>
<w word="yesterday" num="1"/>
<w word="evil" num="1"/>
<w word="took" num="1"/>
<w word="crashed" num="1"/>
<w word="hardware" num="1"/>
<w word="flown" num="1"/>
<w word="strikes" num="1"/>


```
<w word="services" num="1"/>
<w word="combined" num="1"/>
<w word="mastermind" num="1"/>
<w word="vowed" num="1"/>
<w word="destruction" num="1"/>
<w word="back?" num="1"/>
<w word="land" num="1"/>
<w word="age" num="1"/>
<w word="hewlett-packard" num="1"/>
<w word="cover" num="1"/>
<w word="spring-loaded" num="1"/>
<w word="cnnfn" num="1"/>
<w word="1970" num="1"/>
<w word="israelis" num="1"/>
<w word="physics" num="1"/>
</words>
</data>
```

Further Thoughts

An exploration of possibilities and considerations for the project, a group discussion.

This project came about because of an initial impetus to see how one news story overtakes another, in an endless cycle, like waves growing then crashing against a shore, disappearing in favor of the next wave.

The form we present here in the Media Memory project can be considered a limited expression of this initial idea, an example created in the time and with the expertise available to us.

During our frequent group meetings and email-communication, enhancements, changes and alternate options were discussed, and only shot down due to the aforementioned constraints.

Here is a sample of two possibilities that were closely considered using the same format and form as our final presentation, possibilities we may have chosen if the class term had been longer than 10 weeks:

1. To have the display of words go from the past to the present, refreshing each year to reflect the weighted terms from all of the years in the past (from the start date) in a single set of words and a single display. Repeated terms would combine together to form a single weighted frequency, with the most recent terms holding the greatest weight. This would allow changes in the words used in the news for each year to completely transform the display, in some cases erasing what came before.
2. To layer the words for each year, so that instead of the words from years past fitting in around more recent years, they would overlap. Each year's words would fill the entire screen, starting with the current year (2006), the words becoming more transparent the further back their origins. This would preserve the visual weighting of size for every year, while the transparency would reflect the forgetfulness that time brings.