Avoiding Spoilers in Fan Wikis of Episodic Fiction

Shawn M. Jones
Old Dominion University
Norfolk, VA, USA
sjone@cs.odu.edu

Michael L. Nelson
Old Dominion University
Norfolk, VA, USA
mln@cs.odu.edu

ABSTRACT
A variety of fan-based wikis about episodic fiction (e.g., television shows, novels, movies) exist on the World Wide Web. These wikis provide a wealth of information about complex stories, but if readers are behind in their viewing they run the risk of encountering “spoilers” – information that gives away key plot points before the intended time of the show’s writers. Enterprising readers might browse the wiki in a web archive so as to view the page prior to a specific episode date and thereby avoid spoilers. Unfortunately, due to how web archives choose the “best” page, it is still possible to see spoilers (especially in sparse archives).

In this paper we discuss how to use Memento to avoid spoilers. Memento uses TimeGates to determine which best archived page to give back to the user, currently using a minimum distance heuristic. We quantify how this heuristic is inadequate for avoiding spoilers, analyzing data collected from fan wikis and the Internet Archive. We create an algorithm for calculating the probability of encountering a spoiler in a given wiki article. We conduct an experiment with 16 wiki sites for popular television shows. We find that 38% of those pages are unavailable in the Internet Archive. We find that when accessing fan wiki pages in the Internet Archive there is as much as a 66% chance of encountering a spoiler. Using sample access logs from the Internet Archive, we find that 19% of actual requests to the Wayback Machine for pages ended in spoilers. We suggest the use of a different minimum distance heuristic, minpast, for wikis, using the desired datetime as an upper bound.

Categories and Subject Descriptors
H.3.7 [Digital Libraries]: Web Archives, Memento—User Issues

Keywords
Digital Preservation, HTTP, Resource Versioning, Web Archiving, Wikis, Spoilers

1. INTRODUCTION
From How I Met Your Mother to Game of Thrones, fans have created fan-based wikis based on their favorite episodic fiction. For a community of fans these wikis become the focal point for continued discussion and documenting the details of the fictional milieu. The first study on fan wikis was done on the wiki Lostpedia, for the show Lost [15].

Unfortunately, due to the rise in the availability of recorded media, viewing television shows after their air date has become more common, making the use of these wikis difficult for those who have not yet consumed all of the episodes to date, leading to spoilers. Spoilers are defined as pieces of information that user wants to control the time and place of their consumption, preferring to consume them in the order that the author (or director) intended. If these pieces of information are delivered in the wrong order, enjoyment about a movie or television program is damaged [9]. The problem of spoilers has been reported in popular media for years, from such sources as CNN [7] and The New York Times [8].

The Memento Framework [22, 21] can be used to avoid spoilers on the web [11, 12]. Memento allows one to extend content negotiation into the dimension of time, a process called datetime negotiation, allowing a user to choose a date prior to the episode they have not seen and view the web as it looked at that time.

Memento provides several resource types that play a role in datetime negotiation. First, the original resource, also noted as a URI-R, is the page for which we want the past version. In MediaWiki parlance, it is called a topic URI, and refers to the wiki article in its current state. Then we have the memento, from which the Memento Framework gets its name, also noted as URI-M. It is the past version of the page. In MediaWiki parlance, it is called a oldid page. Third, we have the TimeMap, also noted as URI-T, which is a resource associated with the original resource from which a list of mementos for that resource are available. The TimeMap provides a list of URI-Ms and datetimes in a well-defined format, but does not contain any article content. Finally, we have the TimeGate, also noted as URI-G, which is a resource associated with the original resource that provides datetime negotiation. It is the URI to which the user sends a datetime and receives information about which memento (URI-M) is the best match for that datetime. The TimeGate only processes and redirects; it provides no representations itself.

Wikis preserve every revision of a page as mementos, accessible via a series of URI-Ms. The web archive then captures some of those revisions as general mementos, accessible via a different series of URI-Ms. Unfortunately, for a web archive, there are missed updates that are never recorded, so we are unsure of the interval for which any given general memento is valid. For a wiki, we have every revision and no missed updates, so we do know the interval of their valid-
ity. This makes wiki revisions a special case of mementos. For the sake of this paper, to differentiate between the two sources, we use the term revision when referring to a memento saved by a wiki, and the term memento when referring to the more general mementos residing at a web archive. The full discussion of models and durations of validity is outside of the scope of this paper.

Figure 1 shows two timelines. The bottom timeline consists of mementos captured by a web archive. The top line consists of several revisions of a wiki page. From this figure, we see that memento $m_k$ was archived by the web archive at datetime $t_{14}$, which we denote as $m_k @ t_{14}$, or, also more generically $t_{mk}$. Likewise, revision $t_{r_{j-4}} @ t_2$ denotes the time for which wiki revision $r_{j-4}$ was saved. Arrows between the memento line and the revision line show which mementos are captures of which revisions. We denote this as $m_k \equiv r_j$, indicating that $m_k$ is a capture of $r_j$. We see that revisions $r_{j-3}$, $r_{j-2}$, and $r_{j-1}$ are never captured, making them missed updates.

In Figure 2, we add a third timeline for events, showing the pattern observed by Steiner [20] where events inspire wiki revisions to be created. In this case events correspond to television episodes. As seen above, these edits are eventually captured by the web archive. We use the nomenclature $t_{ei}$ to refer to the time of the $i^{th}$ episode. We also use the $e_1$ to refer to the first episode and $e_n$ as the latest (or last) episode.

For this paper, we define the term spoiler naively as any memento that exists after the date desired by the user, regardless of the content of the memento. Figure 3 illustrates this concept using $e_i$ as the episode database, and $r_j$ and $r_j + 1$ as revisions on either side of this event. Based on our definition revision $r_j$ is safe because it exists prior to episode $e_i$ that the user is trying to avoid. It is assumed that revision $r_{j+1}$ contains spoilers because that wiki edit occurred after the $e_i$.

It is this relationship between events and revisions that allow our spoiler solution to work. Fans who edit wiki pages typically have no knowledge of an episode’s content until that episode airs, meaning that revisions containing that information must come after the episode.

In determining the best memento to which the user should be directed, web archives use a minimum distance heuristic. We demonstrate that this heuristic is not useful for avoiding spoilers. Fan wikis are a special case, because they are updated frequently and many of their users want to avoid spoilers. We do not seek to change the Internet Archive’s processes. We can use Memento directly on wikis to avoid spoilers because wikis have access to all revisions [14]. These revisions are mementos in their own right, and because we have all revisions, we can use a different heuristic (minpast), that avoids mementos after the date requested by the user. Thus, by using Memento directly on a wiki, one can avoid spoilers in fan wikis. The Memento MediaWiki Extension provides this functionality for MediaWiki [13], allowing spoiler avoidance for those who install the extension.

As part of this temporal analysis, we will further define the two heuristics under consideration, mindist and minpast. We will also show that there is a 66% probability of encountering a spoiler when using web archives to access prior versions of wikis, because web archives use mindist. In addition to not reliably helping users avoid spoilers, we find that 38% of the pages in our sample are not available in the web archive.

In this paper, we briefly show what others have done to study the spoiler problem, then we discuss what previous studies have been done on wikis. From here we discuss two different TimeGate heuristics and how minpast is preferred over mindist when a user is trying to avoid spoilers. Then we discuss how the mindist heuristic can lead to spoiler areas, where a user selects a datetime prior to an episode they want to avoid, but are still directed into the future. Using these spoiler areas, we then show how one can calculate the probability of encountering a spoiler for a given page.
Armed with these concepts, we show the results of a study performed on 16 fan wikis for popular television shows [12], showing not only that these spoiler areas exist for users, but also the probabilities of encountering spoilers in these sites.

We then discuss the results of a second study on logs from the Wayback Machine, showing that 19% of all requests end up in the future, indicating that the spoiler problem is real, and that the Wayback Machine is not a reliable tool for avoiding spoilers on the web.

2. RELATED WORK

Schirra, Sun, and Bently conducted a study of two-screen viewing while the television show Downton Abbey was airing [19]. Two-screen viewing is a process whereby those watching a television show episode discuss the show on a social media web site, such as Twitter, while the episode is airing. A similar study was conducted by Johns [10]. Both studies discovered that users would use elaborate methods to avoid revealing and encountering spoilers in social media as well as the current versions of web sites.

Because of the phenomenon of spoilers in social media, Boyd-Graber, Glasgow, and Zajac conducted an evaluation of machine learning approaches to find spoilers in social media posts [3]. They used classifiers on multiple sources to determine which posts should be blocked. They mention that spoilers refer to events “later” than the viewer’s knowledge of the current work, suggesting that any machine learning technique used for avoiding spoilers in social media must be smarter than just blocking all posts about a particular topic [6, 9]. Inspired by this work were software packages that block spoilers from a user’s social media feed, such as Spoiler Shield [16] and the Netflix Spoiler Foiler [5].

We are proposing an orthogonal concept relating to fan wikis, not social media. We are also not blocking resources, rather indicating that the fan wiki pages can still be useful resources if past versions of them are accessible to users. Our solution can be combined with a content-based approach, but we are proposing a structural solution that can be combined with content-based solutions in the future.

Almedia, Mozafari, and Cho produced one of the first studies of the behavior of contributors to Wikipedia [3]. The authors discover that there are distinct groups of Wikipedia contributors. They suggest that as the number of articles increases, the contributors’ attention is split among more and more content, resulting in the larger number of revising contributors rather than article creators. This informs our notion of number of edits as a surrogate to the popularity of a page.

Additionally, there has been some effort of preserving wiki pages outside of the Internet Archive. Popitsch, Mosser, and Phillipp have created the UROBE project for archiving wiki representations in a generic format that can then be reconstituted into many other formats for data analysis [17]. Interestingly, they anticipate attaching their process to Memento at some point later in their research so that past versions of their archives can be accessed by datetime.

3. MEMENTO TIMEGATE HEURISTICS

When the user selects a desired datetime prior to the episode they have not yet seen, the TimeGate is what determines which memento they are redirected to. In the case of spoilers, the wrong heuristic can redirect the user to a spoiler even though they requested a datetime prior to the event that would have caused the spoiler.

Memento TimeGates accept two arguments from the user: desired datetime (specified in the Accept-Datetime header) and a URI-R; and they return the best URI-M using some heuristic [2]. RFC 7089 leaves the heuristic of finding the best URI-M up to the implementor, stating that “the exact nature of the selection algorithm is at the server’s discretion but is intended to be consistent” [21]. Figure 4 shows the differences between the mindist and minpast heuristics used for TimeGates.

Mindist (minimum distance) finds the closest memento to the given desired datetime $t_0$. Mindist is best used for web archives which are typically sparse, meaning they may have missed many revisions of a page. In this case, a user would want the closest memento they can get to the date they are requesting because the dates of capture may be wildly distant from one another. Because of the fact that it may choose mementos from a date after the desired datetime, mindist is not a reliable heuristic for avoiding spoilers.

This heuristic is useful in cases where there are few mementos recorded for a web page. Consider an example where only two mementos exist, from 2003 and 2009. If the user wishes to see the page as it looked on 2008, the 2009 (mindist) is likely best. Most web archives are sparse, hence mindist is used to satisfy the majority of use cases. This heuristic is what the Wayback Machine uses, and is not user-configurable.

Minpast, short for minimum distance in the past, finds the closest memento to the desired datetime $t_0$, but without going over $t_0$. Minpast is best used for archives which are abundant with mementos. Ideally, minpast should be used if every revision of a resource has been archived, as with wikis. For wikis, the value of desired datetime $t_0$ corresponds to a revision that actually existed at the time of $t_0$. For web archives that are not abundant, information may be lost because they may not have captured all revisions. Minpast can be used to avoid spoilers. If we select a value for $t_0$ prior to the event we want to avoid, then minpast will not find any mementos after $t_0$. It is best used for wikis where we have access to all revisions because we can definitively state that the memento returned is the page as it existed at $t_0$. 

![Figure 4: Demonstration of the mindist and minpast heuristics; $m_3@t_{10}$ is chosen by mindist whereas $m_2@t_7$ is chosen by minpast](image-url)
4. SPOILER PROBABILITIES

By studying mindist using wiki revisions and the mementos corresponding to them, we can measure the probability of encountering a spoiler for a given wiki page in web archives.

The set of datetimes where the user is redirected to a memento after the episode, even though they chose a datetime prior to the episode is defined as a spoiler area.

The set of datetimes where the user is directed to a spoiler, even though they chose a datetime prior to the episode they are avoiding, and where the web archive has not yet started archiving the resource, is referred to as a pre-archive spoiler area. Figure 5 shows two pre-archive spoiler areas. This spoiler area is created if the user tries to select a datetime prior to episode \( e_3 @ t_{11} \), but the mindist heuristic delivers them to \( m_3 @ t_{14} \equiv r_2 @ t_{13} \), which is after \( e_3 @ t_{11} \). The user intended to avoid spoilers for episode \( e_3 \), but got them nonetheless because the archive’s earliest memento is after the desired datetime.

So, for a pre-archive spoiler area to exist, the following conditions must be present:

1. The TimeGate for the resource uses the mindist heuristic
2. We have access to all revisions of a given resource
3. The Memento-Datetimes times for all revisions of a resource are defined and known
4. Event \( e \) must occur prior to the first memento recorded in the archive
5. Event \( e \) must occur prior to revision \( r_i \) corresponding to the first memento \( m_1 \) (i.e., \( r_i \equiv m_1 \land t_e < t_{r_j} \))

Given episodes \( e_1 \) to \( e_i \), which occur just prior to the first archived revision \( r_j \equiv m_1 \), this gives us the definition of a pre-archive spoiler area for episode \( e_i \) defined by function \( S_\alpha \) over the interval \( t_s \) and ending at finish datetime \( t_f \) produced by Equation (1).

\[
[t_s, t_f] = S_\alpha(e_i) = \begin{cases} (t_{e_1}, t_{e_i}) & \text{if } t_{e_1} < t_{r_j} \\ (0, 0) & \text{otherwise} \end{cases}
\]

Figure 6 shows an archive-extant spoiler area. Let a user select a datetime prior to \( e_i @ t_{11} \). To avoid spoilers, the user needs to be directed to memento \( m_{k-1} \) corresponding to revision \( r_{j-1} \).

Unfortunately, if the user selects a datetime in the area between \( t_\alpha \) and \( e_i @ t_{11} \), mindist will return memento \( m_k @ t_{13} \), even though they chose a datetime prior to \( t_{11} \). Memento \( m_k @ t_{13} \equiv r_j @ t_{12} \), and \( r_j \) exists after the datetime \( t_{11} \) that the user was trying to avoid. Because the user chose a datetime prior to the episode containing spoilers, but the user is redirected to a memento containing spoilers anyway.

Why is this a spoiler area? Remember that mindist finds the minimum distance between the time \( t_\alpha \) specified by the user and any given memento. In Figure 6 we have mementos \( m_{k-1} @ t_5 \) and \( m_k @ t_{13} \). We denote the midpoint between mementos as \( h \) (for halfway). This means that any value \( t_\alpha \) such that \( t_\alpha < t_e < t_{13} \) will produce memento \( m_j \) and any value \( t_\alpha \) such that \( t_\alpha < t_\alpha \) will produce memento \( m_{j-1} \).

So, for an archive-extant spoiler area to exist, the following conditions must be present:

1. The TimeGate for the resource uses the mindist heuristic
2. We have access to all revisions of a given resource
3. The memento-datetimes times for all revisions of a resource are defined and known
4. Event \( e \) must occur between the memento-datetimes of two consecutive mementos \( m_{k-1} \) and \( m_k \) (i.e., \( t_{m_{k-1}} < t_e < t_{m_k} \))
5. Event \( e \) must occur prior to revision \( r_i \) corresponding to memento \( m_j \) (i.e., \( r_j \equiv m_k \land t_e < t_{r_j} \))
6. The midpoint \( t_\alpha \) between \( m_{k-1} \) and \( m_k \) must occur prior to event \( e \) (i.e., \( t_{m_{k-1}} < t_h < t_e < t_{m_k} \))
Figure 7: Example of a potential spoiler zone, stretching from \( t_{e_1} \) to \( t_{e_n} \)

Figure 8: Example of a spoiler area (light red area) for episode \( e_i \) inside potential spoiler zone (dotted red rectangle), stretching from \( t_{e_1} \) to \( t_{e_n} \)

Given consecutive mementos \( m_{k-1} \) and \( m_k \), the midpoint \( t_h \) between them, and revision \( r_j \equiv m_k \), this gives us the definition of an archive-extant spoiler area defined by function \( S_{b} \) over the interval beginning at start datetime \( t_b \) and ending at finish datetime \( t_f \) produced by Equation (2).

\[
[t_b, t_f] = S_{b}(c) = \begin{cases} 
(t_h, t_e) & \text{if } t_h < t_e < t_{e_1} \\
& \text{and } r_j \equiv m_k \land \text{mk}_{k-1} < t_h < \text{mk}_{k+1} \\
(t_m, 0) & \text{otherwise}
\end{cases}
\]  

(2)

So, how does one handle multiple episodes? What does that mean for our spoiler areas? For a given resource, using mindist, what is the chance of attempting web time travel with Memento and getting a spoiler?

First we define a potential spoiler zone across the length of the series we are looking at. The start datetime of the potential spoiler zone is \( t_{e_1} \), the datetime of the first episode. The end datetime of our potential spoiler zone is \( t_{e_n} \), the datetime of the last (or latest) episode. We assume that a user searching for datetimes prior to the first event \( e_1 \) should get no spoilers, so that is the lower bound. We also assume that no additional spoilers can be revealed after the last event \( e_n \). This provides a single area in which we can determine the probability of getting a spoiler for a single episode in the series. Figure 4 shows an example of such a zone.

Figure 5 shows a spoiler area \( (t_4 \text{ to } t_5) \) inside a potential spoiler zone \( (t_{e_1} \text{ to } t_{e_n}) \). Consider randomly choosing a desired datetime within this zone. What is the probability of landing inside the spoiler area for given episode \( c \)?

Probability is defined as the number of times something can occur divided by the total number of outcomes \([23]\). The smallest unit of datetime on the web is the second. We cannot gain more precision over time due to the fact that HTTP headers (and hence Memento-Datetimes) use the second as the smallest unit. Consider iterating through every second between \( e_1 \) and \( e_n \), incrementing the value of counter \( s \) for each second that falls within a spoiler area. If we let \( c \) be the number of seconds between \( e_1 \) and \( e_n \), then the probability of encountering a spoiler is shown by equation (3).

\[
Pr(\text{spoiler}) = \frac{s}{c}
\]  

(3)

Once we have determined the probability of encountering a spoiler for a resource within the Internet Archive, we can then use that probability to compare that resource to others. In this way we can determine how safe a given URI is for users who want to avoid spoilers using the Wayback Machine or a Memento TimeGate that uses the mindist heuristic.

5. MEASURING SPOILER PROBABILITY IN POPULAR WIKIS

We selected 16 fan wikis based on television shows for our experiment. Table 1 shows some of details for each fan wiki. Each television show selected has had at least two seasons and a currently active wiki. House of Cards was chosen because an entire season is released on Netflix in a single day, making it different from networked television shows. Lost was chosen because its wiki, Lostpedia, has undergone academic study \([15]\), and is the oldest and largest fan wiki under consideration. We used a process, simplified in Algorithm 1, to process each wiki and identify the spoiler areas created by mindist. Episode dates were supplied by epguides.com.

Utilizing this method, we computed additional statistics based on the revisions, mementos, the memento-revision mapping, and the spoiler areas.

Out of the 40,868 wiki pages processed for this experiment, we discovered that many of them were wiki redirects. Redirects are used to deal with articles that can be referred to by multiple names. Sometimes wiki editors may not know the real name of an introduced character until much later, and will use a redirect from the old name to the new. Sometimes wiki editors will create pages not knowing that one already exists, leaving future editors to create a redirect now that they know that a new page title was desired. Because of the number of redirects that contained only a single revision and only a single memento, we removed the redirects from consideration for calculation of spoiler areas and other statistics. This removed 16,394 pages from consideration, leaving us with 24,474 pages to process.

The wiki XML exports were downloaded at a different time than the TimeMaps for those wiki pages. To overcome this
## Table 1: Fan wikis used in the spoiler areas experiment

| Television Show (Network) | Wiki URI | # of Pages | $t_{r_1}$ | $t_{r_2}$ | % of pages in Internet Archive |
|--------------------------|----------|------------|-----------|-----------|-------------------------------|
| the Big Bang Theory (CBS) | bigbangtheory.wikia.com | 1120 | 2007-12-14 | 2007-09-24 | 68.5% |
| Boardwalk Empire (HBO) | boardwalkempire.wikia.com | 2091 | 2010-03-18 | 2010-08-23 | 80.6% |
| Breaking Bad (AMC) | breakingbad.wikia.com | 998 | 2005-09-20 | 2008-04-20 | 16.0% |
| Continuum (SyFy) | continuum.wikia.com | 258 | 2012-11-15 | 2012-05-27 | 68.8% |
| Downton Abbey (BBC) | downtonabbey.wikia.com | 784 | 2010-10-04 | 2010-09-26 | 53.1% |
| Game of Thrones (HBO) | gameofthrones.wikia.com | 3144 | 2010-06-24 | 2011-04-17 | 75.8% |
| Grimm (NBC) | grimm.wikia.com | 1581 | 2010-04-14 | 2011-10-28 | 57.5% |
| House of Cards (Netflix) | house-of-cards.wikia.com | 251 | 2013-01-11 | 2013-02-01 | 97.2% |
| How I Met Your Mother (CBS) | how-i-met-your-mother.wikia.com | 1709 | 2008-07-21 | 2005-09-19 | 58.7% |
| Lost (ABC) | lostpedia.wikia.com | 18790 | 2005-09-22 | 2004-09-22 | 39.1% |
| Mad Men (AMC) | madmen.wikia.com | 632 | 2009-07-25 | 2007-06-05 | 85.6% |
| NCIS (CBS) | ncis.wikia.com | 6345 | 2009-09-25 | 2004-09-23 | 93.2% |
| Once Upon A Time (ABC) | onceuponatime.wikia.com | 1470 | 2011-08-09 | 2011-10-23 | 79.9% |
| Scandal (ABC) | scandal.wikia.com | 331 | 2011-06-07 | 2012-04-05 | 82.8% |
| True Blood (HBO) | trueblood.wikia.com | 1838 | 2008-10-06 | 2008-09-07 | 74.1% |
| White Collar (USA) | whitecollar.wikia.com | 506 | 2009-10-30 | 2009-10-23 | 79.1% |

## Table 2: Spoiler probabilities for most popular pages within each fan wiki

| Wiki | Page Name | Probability of Spoiler | # of Spoiler Areas | # of Revisions | # of Mementos |
|------|-----------|------------------------|--------------------|----------------|--------------|
| bigbangtheory | Sheldon Cooper | 0.31 | 69 | 1958 | 30 |
| boardwalkempire | Nucky Thompson | 0.15 | 31 | 290 | 15 |
| breakingbad | Walter White | 0.43 | 40 | 882 | 20 |
| continuum | Keira Cameron | 0.54 | 21 | 104 | 5 |
| downtonabbey | Sybil Branson | 0.42 | 23 | 580 | 3 |
| gameofthrones | Daenerys Targaryen | 0.16 | 24 | 708 | 29 |
| grimm | Nick Burkhardt | 0.39 | 30 | 795 | 5 |
| house-of-cards | Frank Underwood | 0.9 | 13 | 380 | 3 |
| how-i-met-your-mother | Barney Stinson | 0.55 | 120 | 588 | 13 |
| lostpedia | Kate Austen | 0.67 | 94 | 3531 | 27 |
| madmen | Mad Men Wiki | 0.22 | 36 | 250 | 85 |
| ncis | Abigail Sciuto | 0.67 | 182 | 404 | 11 |
| onceuponatime | Emma Swan | 0.36 | 34 | 1210 | 11 |
| scandal | Main Page | 0.60 | 31 | 250 | 14 |
| trueblood | Eric Northman | 0.28 | 47 | 931 | 14 |
| whitecollar | Neal Caffrey | 0.29 | 38 | 199 | 8 |

## Table 3: Statistics for each fan wiki

| Wiki | Probability of Spoiler | Revisions/Day | Mementos/Day |
|------|------------------------|---------------|--------------|
| Mean | std dev | Rel Err | Mean | std dev | Rel Err | Mean | std dev | Rel Err |
| bigbangtheory | 0.667 | 0.160 | 0.0116 | 0.0506 | 0.0698 | 0.0369 | 0.0035 | 0.0034 | 0.0488 |
| boardwalkempire | 0.417 | 0.170 | 0.0160 | 0.0102 | 0.0185 | 0.0718 | 0.0022 | 0.0020 | 0.0452 |
| breakingbad | 0.746 | 0.205 | 0.0127 | 0.0185 | 0.0351 | 0.0872 | 0.0032 | 0.0032 | 0.0459 |
| continuum | 0.394 | 0.177 | 0.0471 | 0.0317 | 0.0250 | 0.0829 | 0.0051 | 0.0023 | 0.0479 |
| downtonabbey | 0.585 | 0.174 | 0.0196 | 0.0374 | 0.0636 | 0.1124 | 0.0020 | 0.0013 | 0.0419 |
| gameofthrones | 0.473 | 0.248 | 0.0122 | 0.0425 | 0.0652 | 0.0356 | 0.0041 | 0.0049 | 0.0279 |
| grimm | 0.479 | 0.175 | 0.0201 | 0.0700 | 0.0857 | 0.0672 | 0.0027 | 0.0015 | 0.0306 |
| house-of-cards | 0.606 | 0.035 | 0.6705 | 0.0772 | 0.1364 | 0.2082 | 0.0075 | 0.0044 | 0.0687 |
| how-i-met-your-mother | 0.741 | 0.100 | 0.0046 | 0.0163 | 0.0220 | 0.0463 | 0.0014 | 0.0010 | 0.0263 |
| lostpedia | 0.768 | 0.163 | 0.0027 | 0.0391 | 0.1083 | 0.0348 | 0.0040 | 0.0055 | 0.0173 |
| madmen | 0.590 | 0.144 | 0.0153 | 0.0049 | 0.0076 | 0.0764 | 0.0014 | 0.0021 | 0.0768 |
| ncis | 0.818 | 0.107 | 0.0041 | 0.0073 | 0.0097 | 0.0413 | 0.0009 | 0.0008 | 0.0279 |
| onceuponatime | 0.516 | 0.163 | 0.0132 | 0.1271 | 0.1327 | 0.0437 | 0.0037 | 0.0025 | 0.0281 |
| scandal | 0.591 | 0.165 | 0.0269 | 0.0418 | 0.0484 | 0.1120 | 0.0030 | 0.0019 | 0.0608 |
| trueblood | 0.517 | 0.162 | 0.0106 | 0.0210 | 0.0410 | 0.0658 | 0.0016 | 0.0016 | 0.0345 |
| whitecollar | 0.390 | 0.250 | 0.0500 | 0.0117 | 0.0147 | 0.0986 | 0.0019 | 0.0015 | 0.0699 |
| Overall | 0.659 | 0.226 | 0.0029 | 0.0362 | 0.0871 | 0.0200 | 0.0032 | 0.0044 | 0.0114 |
Algorithm 1: Algorithm for spoiler probability experiment

1. \( \text{FindSpoilerAreasInWikis}(\text{episodeList}, \text{wikiURI}) \)
2. \( \text{episodeTimes} = \text{getEpisodeTimes}(\text{episodeList}) \)
3. \( \text{wikiTitles} = \text{getPageTitles}(\text{wikiURI}) \)
4. \( \text{for each} \ \text{title} \in \text{wikiTitles} \)
5. \( \text{wikidump} = \text{fetchXMLdump}(\text{title}, \text{wikiURI}) \)
6. \( \text{revisions} = \text{extractRevisionTimes}(\text{wikidump}) \)
7. \( \text{timemapURI} = \text{makeTMURI}(\text{wikiURI}, \text{title}) \)
8. \( \text{timemap} = \text{fetchTimeMap}(\text{timemapURI}) \)
9. \( \text{mementos} = \text{extractMementoTimes}(\text{timemap}) \)
10. \( \text{mementoRevisionMap} = \text{mapRevsToMems}(\text{revisions}, \text{mementos}) \)
11. \( \text{for each} \ \text{episode} \in \text{episodeTimes} \)
12. \( \text{paSpoilerArea} = \text{S}_a(\text{episode}, \text{mementoRevisionMap}) \)
13. \( \text{aeSpoilerArea} = \text{S}_b(\text{episode}, \text{mementoRevisionMap}) \)
14. \( \text{spoilerAreaList.append}(\text{paSpoilerArea}) \)
15. \( \text{spoilerAreaList.append}(\text{aeSpoilerArea}) \)
16. \( \text{return} \ \text{wikiPageSpoilerMap} \)

Of the 24,474 pages processed, only 15,119 pages actually had TimeMaps at the Internet Archive at the time the wiki exports were extracted. This means that roughly 38% of the pages under consideration were not available in the Internet Archive.

Figure 9 shows our spoiler area graph for the page with the most revisions in our entire dataset, a page from Lostpedia about the character Kate Austen. Each spoiler area is shown in red using an alpha channel that gives it some degree of transparency. When these transparent red areas stack up, of course the red gets darker, so we cannot reliably see all of the pre-archive spoiler areas that exist prior to the first memento. The probability of encountering a spoiler for Kate’s page is 67%, calculated by Equation (3). Because this page only has a few mementos around 2009 and then a long break for the Internet Archive until 2011, there are a few archive-extant spoiler areas, also shown in red, both around the 2009 mark. We also see some archive-extant spoiler areas, and also after the memento halfway mark in 2010.

Figure 10 shows spoiler areas for the page about the Big Bang Theory character Sheldon Cooper. The Internet Archive is more aggressive at archiving in 2008 than it was during the run of the show Lost (starting 2004), so there are only 8 pre-archive spoiler areas for this page, compared to Kate’s 86. There are, however, 61 archive-extant spoiler areas, compared with Kate’s 8. Sheldon’s page has a spoiler probability of only 31%. We can see the clusters of points indicating each episode on the events timeline. Because television show seasons occur during portions of the year, we can see the seasons, and partial seasons, for Big Bang Theory on the top. Even though Sheldon’s page contains quite a few spoiler ar-

Figure 9: Spoiler areas for the most popular page (3,531 revisions) in our data set

Figure 10: Spoiler areas for the most popular page (1,958 revisions) in the Big Bang Theory Wiki

Figure 11: Spoiler areas for the most popular page (768 revisions) in the Game of Thrones Wiki
eas after the second season, there appears to be a block of
time before the third season where one is safe to browse this
page and avoid spoilers.

Figure 11 provides another example of a more current show,
using a page from the *Game of Thrones Wiki*.

Table 2 contains statistics for the most popular page in each
of the wikis that we have surveyed, where popularity is
determined by the number of page revisions generated. Seeing
as these wikis are authored by fans, readers familiar with
many of these television shows will not be surprised that
most of the popular pages are main characters. The table
also lists the number of spoiler areas, revisions, and mementos,
showing how there is not a simple relationship between
these values that indicate the probability of encountering a
spoiler.

Of particular interest is the television show, *House of Cards*. Because it releases an entire season of episodes at one time,
our model breaks down. We count 13 pre-archive spoiler ar-
Eas for the first season, and then no archive-extant spoiler ar-
Eas. The pre-archive spoiler areas have no size due to the fact
that all of them begin and end at the same time. This leads
to a 0% chance of encountering a spoiler in this wiki, seeing
as each season is released like a 13-hour movie rather than
on a weekly basis. In this case, time is not able to differenti-
ate between individual episodes because $t_{e_1} = t_{e_2} = \ldots t_{e_{13}}$. It requires a new dimension in order to order otherwise si-
multaneous events. A different situation exists with another
Netflix series, *Arrested Development*, in which all episodes
for a season are released at once, but the episodes do not
need to be viewed in any particular order, making it diffi-
cult to identify when spoilers would occur.

Table 3 shows the statistics for each fan wiki. We see a
mean overall spoiler probability of 66%. We also see that
the number of mementos per day is an order of magnitude
smaller than the number of revisions per day.
Figure 12 shows the probability distribution of encountering spoilers in these wiki pages. Figure 13 shows a cumulative distribution function of spoiler probabilities for all wikis within the data set. Here we see that the spoiler probability exists, in some form, for most of the pages.

Figure 14 shows the number of missed updates encountered for each datetime over the history of all pages in the wiki. The Y-axis represents each URI in the data set. The X-axis is time. Lighter colors indicate fewer missed updates on that day. Of interest are the vertical lines seen throughout the visualization. The datetimes for these lines correspond to changes in policy at the Internet Archive. In 2009 and in late 2011, the Internet Archive reduced its quarantine period for archiving of new pages. In October of 2013, the Internet Archive published the Save Page Now feature [18], leading to fewer missed updates after that point.

Figure 15 shows the number of redundant mementos created for each datetime over the history of all pages in the wiki. Just as with Figure 14, the Y-axis represents each URI and the X-axis is time. As expected, the number of redundant mementos increases as the Internet Archive becomes more aggressive about archiving web pages.

6. MEASURING NAIVE SPOILERS IN WAYBACK MACHINE LOGS

Research has already been done by Ainsworth in how much drift exists within the web archive [1]. That study indicates that the Wayback Machine uses a sliding target policy. This means that each request is in some way based on the datetime of the last request, resulting in a user ending up in a much different datetime than they had originally started. The Wayback Machine still uses the mindset heuristic to determine which memento to deliver to a user, but it changes the desired datetime $t_a$ based on the datetime of the memento from the last request.

Contrary to this, Memento uses a sticky target policy, allowing a user to fix the datetime $t_a$ throughout their browsing session. While the sparsity of the archives introduces some small drift with the sticky target policy, it is constrained by the datetime remaining constant in each request. That drift is introduced only by the mindset heuristic rather than the sliding behavior of the Wayback Machine.

We are concerned about whether or not the user ended up in the future of where they intended. We want to know if they encountered a spoiler when using the Wayback Machine. We conducted a study using anonymized Wayback Machine logs spanning January 1, 2011 through March 10, 2011 and August 1, 2011 through March 26, 2012.

The logs from the Wayback Machine are in Apache common log format. Using the referrer for each request, we can track where the user came from and determine where they ended up. Fortunately for us, we can infer the desired datetime (referred to as $t_r$ previously) and the memento-datetime from the URIs themselves. The Internet Archive allows access to all mementos using a standard URI format and the datetime is embedded in the URI. For the URI visited by the user, this datetime indicates the memento-datetime. For the referrer URI, this datetime indicates their desired datetime.

Why do we say that we can infer the desired datetime? Without interviewing the visitors to the Wayback Machine, it is impossible to determine intent. The fact that the logs are anonymized makes this completely impossible. We are making the assumption that some of the users receiving these responses intended to receive responses on the date that they started at, not the date delivered by the drift caused by the mindset heuristic.

From these logs we can determine the inferred desired datetime from the referrer URI and the memento-datetime from the visited URI. Using this information, we can download the wiki exports, as in the previous experiment, and determine if the page revision recorded by the web archive exists in the future of the desired datetime.

All requests for archived pages from wikia.com were extracted from the logs, resulting in 1,180,759 requests. Of those requests, we removed all requests for images, JavaScript, style sheets, supporting wiki pages (such as Template, Category, and Special pages), and advertisements. This left us with 62,227 requests to review.

For those remaining wikia.com pages, we downloaded the wiki export files, as done in the previous experiment, mapped the visited URI to the request that it had archived, and compared the datetime of that revision with the inferred desired datetime. We use $t_a$ to represent the inferred desired time, and $t_r$ to represent the datetime of the wiki revision matching the visited URI in the Wayback Machine.

Each response can be split into three categories in terms of spoilers: (1) spoiler - $t_a < t_r$; (2) safe - $t_a \geq t_r$; (3) indeterminate - either the datetime for the revision or the referrer was not able to be determined, likely because the article or whole wiki was moved or no longer exists, or because of 503 HTTP status codes due to the size of the export file.

This process, shown in Algorithm 2, determines how many requests are either spoiler, safe, or indeterminate for each log file. Indeterminate entries make up the bulk of the data collected, but offer no meaningful insight into the spoiler problem, and are thus discarded. From this study we found that roughly 19% of these requests to the Wayback Machine result in spoilers.

```
Algorithm 2: Algorithm for Detecting spoilers in Internet Archive Logs

1: for each visitorID, visitedURI, referrer \in log file
2: \hspace{1em} $t_m = $getDate(visitedURI)
3: \hspace{1em} $t_r = $getDate(referrer)
4: \hspace{1em} wikidump = fetchXMLDump(title, wikiURI)
5: \hspace{1em} revisions = extractRevisionTimes(wikidump)
6: \hspace{1em} $t_r = $getRevMatchingMemento($t_m, revisions)
7: \hspace{1em} spoiler = INDETERMINATE
8: if rev is not NULL
9: \hspace{1em} \hspace{1em} spoiler = ($t_a < t_r$)
10: \hspace{1em} \hspace{1em} print(visitorID + ", " + spoiler)
```
7. CONCLUSIONS
We have introduced the notion of different heuristics for use with Memento TimeGates. We have shown that the mindist heuristic, while useful for sparse archives, is not reliably effective for users trying to avoid spoilers with Memento. We have also proposed minpast as a superior choice for wikis, who have access to every revision.

We have shown that roughly 38% of the pages under consideration were not available in the Internet Archive. We also found that, for the wiki sites under consideration, there is a mean 66% probability that one will end up with a spoiler if they use TimeGates supporting the mindist heuristic. Also, from our sample logs from the Wayback Machine, 19% of requests to wikia.com end in spoilers. This presents a problem for episodic fiction fans trying to use the Wayback Machine, or the Internet Archive through Memento, to avoid spoilers. This further demonstrates that using Memento directly on wikis, using minpast, is better for avoiding spoilers.

8. ACKNOWLEDGMENTS
This work made possible in part by a grant from the Andrew Mellon Foundation. We are grateful to the Internet Archive for their continued support of Memento. Memento is a joint project between the Los Alamos National Laboratory Research Library and Old Dominion University.

9. REFERENCES
[1] S. G. Ainsworth and M. L. Nelson. Evaluating sliding and sticky target policies by measuring temporal drift in acyclic walks through a web archive. In JCDL, pages 39 – 48, 2013.
[2] S. G. Ainsworth, M. L. Nelson, and H. Van de Sompel. A Framework for Evaluation of Composite Memento Temporal Coherence. Technical Report arXiv:1402.0928, Old Dominion University, Feb. 2014.
[3] R. Almeida, B. Mozafari, and J. Cho. On the Evolution of Wikipedia. In JICWSM, 2007.
[4] J. Boyd-Graber, K. Glasgow, and J. S. Zajac. Spoiler Alert : Machine Learning Approaches to Detect Social Media Posts with Revelatory Information. In ASIST, volume 50, pages 1–9, 2013.
[5] J. Denham. Netflix releases House of Cards ‘Spoiler Foiler’ for Twitter users. http://www.independent.co.uk/arts-entertainment/tv/news/netflix-releases-house-of-cards-spoiler-foiler-for-twitter-users-9136324.html February 2014.
[6] J. Golbeck. The Twitter Mute Button: A Web Filtering Challenge. In SIGCHI, pages 2755–2758, 2012.
[7] B. D. Gross. Spoiler alert! Negotiating social media in the DVR age. http://www.cnn.com/2014/02/25/tech/social-media/spoilers-social-media/index.html Feb. 2014.
[8] H. Hart. Spoiler Wars Heat Up as Lost Returns. New York Times. Jan. 2009.
[9] S. Jeon, S. Kim, and H. Yu. Don’t Be Spoiled by Your Friends: Spoiler Detection in TV Program Tweets. In Seventh International AAAI Conference on Weblogs and Social Media, pages 681–684, 2013.
[10] M. D. Johns. Two Screen Viewing and Social Relationships: Exploring the invisible backchannel of TV viewing. In CATTAc, pages 333–343, 2012.
[11] S. M. Jones. 2013-12-18: Avoiding Spoilers with the Memento Mediawiki Extension. http://ws-dl.blogspot.com/2013/12/2013-12-18-avoiding-spoilers-with.html December 2013.
[12] S. M. Jones. Avoiding Spoilers on MediaWiki Fan Sites Using Memento. Master’s thesis, Old Dominion University, 2015.
[13] S. M. Jones, M. L. Nelson, H. Shankar, and H. Van de Sompel. Bringing Web Time Travel to MediaWiki: An Assessment of the Memento MediaWiki Extension. Technical Report arXiv:1406.3, Old Dominion University, June 2014.
[14] B. Leuf and W. Cunningham. The Wiki Way: Quick Collaboration on the Web, 2001.
[15] J. Mittell. Sites of participation: Wiki fandom and the case of Lostpedia. In TWC, September 2009.
[16] Picking Carrots LLC. Spoiler Shield. http://www.spoilershield.com October 2014.
[17] N. Popitsch, R. Mosser, and W. Philipp. Urobe: A prototype for wiki preservation. In IPRES 2010, September 2010.
[18] A. Rossi. Fixing Broken Links on the Internet. https://blog.archive.org/2013/10/25/fixing-broken-links/ October 2013.
[19] S. Schirra, H. Sun, and F. Bentley. Together alone. In SIGCHI, pages 2441–2450, 2014.
[20] T. Steiner, S. van Hooland, and E. Summers. MJ No More: Using Concurrent Wikipedia Edit Spikes with Social Network Plausibility Checks for Breaking News Detection. In WWW, pages 791–794, 2013.
[21] H. Van de Sompel, M. L. Nelson, and R. Sanderson. HTTP Framework for Time-Based Access to Resource States – Memento, Internet RFC 7089. http://tools.ietf.org/rfc/rfc7089.txt Dec. 2013. Accessed: 2014-05-03.
[22] H. Van de Sompel, M. L. Nelson, R. Sanderson, L. Balakireva, S. Ainsworth, and H. Shankar. Memento: Time travel for the web, Technical Report arXiv:0911.1112, Los Alamos National Laboratories and Old Dominion University, 2009.
[23] A. Yaspan. Essentials of Probability. 1968.
APPENDIX

Figure 16: Spoiler areas for the most popular page in *Lostpedia*[^1]

Figure 17: Spoiler areas for the page in *the Big Bang Theory Wiki* that contains the most revisions[^2]

[^1]: [http://lostpedia.wikia.com/wiki/Kate_Austen](http://lostpedia.wikia.com/wiki/Kate_Austen)
[^2]: [http://bigbangtheory.wikia.com/wiki/Sheldon_Cooper](http://bigbangtheory.wikia.com/wiki/Sheldon_Cooper)
Figure 18: Spoiler areas for the page in the *Boardwalk Empire Wiki* that contains the most revisions

Figure 19: Spoiler areas for the page in the *Breaking Bad Wiki* that contains the most revisions

[^http://boardwalkempire.wikia.com/wiki/Nucky_Thompson]
[^http://breakingbad.wikia.com/wiki/Walter_White]
Figure 20: Spoiler areas for the page in the *Continuum Wiki* that contains the most revisions.

Figure 21: Spoiler areas for the page in the *Downton Abbey Wiki* that contains the most revisions.
Figure 22: Spoiler areas for the most popular page in the *Game of Thrones Wiki*.

Figure 23: Spoiler areas for the page in the *Grimm Wiki* that contains the most revisions.
Figure 24: Spoiler areas for the most popular page in the *House of Cards Wiki*.

Figure 25: Spoiler areas for the most popular page in the *How I Met Your Mother Wiki*.
Figure 26: Spoiler areas for the page in the *Mad Men Wiki* that contains the most revisions.\(^{11}\)

Figure 27: Spoiler areas for the page in the *NCIS Database* that contains the most revisions.\(^{12}\)

\(^{11}\)http://madmen.wikia.com/wiki/Mad_Men_Wiki

\(^{12}\)http://ncis.wikia.com/wiki/Abigail_Sciuto
Figure 28: Spoiler areas for the page in the Once Upon A Time Wiki that contains the most revisions

Figure 29: Spoiler areas for the page in the Scandal Wiki that contains the most revisions

http://onceuponatime.wikia.com/wiki/Emma_Swan/Gallery
http://scandal.wikia.com/wiki/Main_Page
Figure 30: Spoiler areas for the page in the True Blood Wiki that contains the most revisions.

Figure 31: Spoiler areas for the page in the White Collar Wiki that contains the most revisions.

15 http://trueblood.wikia.com/wiki/Eric_Northman
16 http://whitecollar.wikia.com/wiki/Neal_Caffrey