# Optimizing the Recency-Relevancy Trade-off in Online News Recommendations

Abhijnan Chakraborty[*†] Saptarshi Ghosh[*○] Niloy Ganguly[*] Krishna P. Gummadi[†]

[†]Max Planck Institute for Software Systems, Germany
[*]Indian Institute of Technology Kharagpur, India
[○]Indian Institute of Engineering Science and Technology Shibpur, India

## ABSTRACT

Online news media sites are emerging as the primary source of news for a large number of users. The selection of 'front-page' stories on these media sites usually takes into consideration several crowdsourced popularity metrics, such as number of views or shares by the readers. In this work, we focus on automatically recommending front-page stories in such media websites. When recommending news stories, there are two basic metrics of interest – recency and relevancy. Ideally, recommender systems should recommend the most relevant stories soon after they are published. However, the relevancy of a story only becomes evident as the story ages, thereby creating a tension between recency and relevancy. A systematic analysis of popular recommendation strategies in use today reveals that they lead to poor trade-offs between recency and relevancy in practice. So, in this paper, we propose a new recommendation strategy (called *Highest Future-Impact*) which attempts to optimize on both the axes. To implement our proposed strategy in practice, we develop an optimization framework combining the predicted future-impact of the stories with the uncertainties in the predictions. Evaluations over three real-world news datasets show that our implementation achieves good performance trade-offs between recency and relevancy.

## Keywords

Front-page News Selection; Non-personalized Recommender Systems; Recency; Relevancy

## 1. INTRODUCTION

A recent Pew survey [1] found that online news media sites, such as those operated by traditional media organizations like The New York Times (`nytimes.com`) and The Guardian (`theguardian.com`), or social news media organizations like Facebook and Twitter, have emerged as the primary source of news for a large and rapidly growing fraction of people world-wide. To ensure the return of audience to their sites at regular interval[1], online media sites constantly update news stories on their front (home) page round the clock (24/7). For instance, `nytimes.com` (NYTimes) updates its spotlighted homepage stories as frequently as once every 15 minutes, while social media sites like Twitter and Facebook update their *trending stories* every 5 minutes [2]. Such constant updates are hard to manage *manually* using only human editors. As a result, there is a growing interest in *automated methods* for recommending front-page news stories. In this paper, our focus is on this task of *automatically recommending (or selecting) front-page stories* on online news media sites.

The recommendation tasks we consider are *global recommendations* that are the same for all users (or viewers) of the site. For instance, the same front-page stories are shown to all news readers visiting a media site, where *personalized factors* (e.g., interests of individual readers) are *not* used for selecting/recommending them. As we show in the subsequent sections, recommending such stories is surprisingly tricky, even when they are *not* personalized.

When globally recommending news stories, there are two basic metrics of interest - *recency* and *relevancy*. By recency, we refer to the age of a story, i.e., the time since its publication. By relevancy, we refer to the importance or the impact of a story, which is often estimated through *crowd-driven measures of popularity*, such as the number of people who read or liked the story. Ideally, recommender systems should select the most relevant stories while they are still recent, i.e., soon after they are published.

Estimating relevancy for global recommendations poses a different set of challenges, compared to judging the relevancy of personalized recommendations. While relevancy of a news story for a particular user can be judged based on the past actions of the user, it is hard to assess the global impact of a story right after its publication (even for human editors, and particularly for automated recommendations). The impact of a story only becomes evident as the story ages, which creates a fundamental tension between selecting recent stories with uncertain relevancy or choosing highly relevant stories that are not recently published. Our goal is to understand and optimize for this *recency-relevancy trade-off* when recommending news stories.

We begin by analyzing the recency-relevancy trade-offs offered by the recommendation strategies which are in use today. Specifically, we investigate (i) *Recent-Impact*-based rec-

[1]Similar to most online websites, many online news media sites are also predominantly funded by their users watching advertisements on their sites.


*WWW 2017,* April 3 – 7, 2017, Perth, Australia.
ACM 978-1-4503-4913-0/17/04.
http://dx.doi.org/10.1145/3038912.3052656

ommendations (used in NYTimes recommendations), where stories which were most popular in the latest time interval are selected, and (ii) *Rising-Impact*-based recommendations (used in Twitter trending topics), where stories that received the sharpest spike in popularity in the most recent time interval, compared to the previous time interval, are chosen. These strategies are based on two key assumptions about popularity life-cycles of news stories (e.g., how the number of views a story receives evolves over time): (i) popularity life-cycles of all stories are somewhat similarly skewed (otherwise a low-impact story that receives all its views in a short time interval would be selected over a high-impact story that steadily accumulates views over a longer time interval), and (ii) news stories achieve their peak recent-popularity or rising-popularity early in their life-cycles (allowing them to be chosen soon after their publications). Our analysis, using real-world news stories datasets, shows that these assumptions do not hold quite frequently, leading to poor trade-offs between recency and relevancy in practice.

Next, we evaluate a simple, but previously unexplored, strategy called *Future-Impact*-based recommendations, where stories are selected based on how many views they are expected to receive in the future (and *not* in the past). Intuitively, future-impact of a story captures the extent to which the story is likely to be discussed in the future, and journalism studies have argued that it is a useful metric for selecting news stories in its own right [3]. Additionally, two properties of the future-impact metric help achieve better trade-offs between recency and relevancy of the recommended stories: (i) a high-impact story has higher future-impact than a low-impact story, and (ii) news stories have highest future-impact shortly after they are published.

Finally, we tackle the technical challenges related to the deployment of the future-impact strategy. The basic idea is to *predict* the future-impact of a story at time $t$, based on the observed impact of the story till time $t$. The key difficulty lies in estimating and accounting for the uncertainty in future-impact predictions, which can be large soon after the story's publication, but decreases over time. We test two different prediction models: using least squares regression, and decision trees (with boosting), and estimate uncertainties in future-impact predictions in both models. We then propose a strategy that selects a set of high future-impact news stories with minimal cumulative uncertainties in their future-impact predictions. Evaluation over real-world news datasets shows that our strategy achieves good performance trade-offs between recency and relevancy.

In summary, the paper makes the following three contributions: (i) we analyze the recency-relevancy trade-offs achieved by current news recommendation strategies and show them to be sub-optimal, (ii) we propose a simple yet previously overlooked strategy that selects stories based on their future-impact, and show that it has the potential to achieve better recency-relevancy trade-offs than current strategies, and (iii) we propose a practical implementation of future-impact based recommendation strategy, tackling the challenge of estimating and accounting for uncertainties in predicting future-impact. Evaluations over three real-world datasets show that our implementation achieves good performance in recommending news stories.

# 2. DATASETS USED

In this work, we used three datasets representing different interaction patterns between news stories and their readers. The first two datasets contain the viewing patterns of different news stories, and the third dataset is regarding the news sharing patterns on Twitter.

**(i) Yahoo! News Dataset:** We used the user click log made publicly available by Yahoo! [4]. Specifically, the R6B dataset contains a fraction of the user click information for 652 news stories displayed on the 'Today Module' on Yahoo!'s front page during the consecutive 15-day period from October 2 to October 16, 2011. The dataset contains information regarding 28,041,015 user visits to the 'Today Module' during this period. During each visit, a story was chosen uniformly at random following the method developed by Li et al. [5], and shown to the user. The click log contains the information about whether the user clicked on the story or not. As the data contains the timestamps of the user visits, for each story, we extracted the sequence of clicks over time. However, one limitation of the dataset is that the stories are anonymized, and no additional information regarding the stories is provided. In absence of any information on the stories, we considered the timestamp of the first click to a story as the publish time of that story.

**(ii) CLEF NewsREEL Dataset:** CLEF NewsREEL provides an evaluation platform to compare different news recommender systems' performance in online and offline settings [6]. As part of the offline evaluation setting, CLEF NewsREEL 2016 shared the user click information recorded for 244,448 news stories from three German news domains: *tagesspiegel.de*, *sport1.de*, and *gulli.com* during July 1, 2014 to August 31, 2014. The data contains information regarding the user-news interactions during this 2 month period. Similar to Yahoo! news data, we extracted the chronological sequence of clicks for every news story. Additionally, we obtained the publish times for the stories in the dataset.

**(iii) NYTimes Dataset:** Apart from using the user click information, we also gathered how stories published by NYTimes are shared on Twitter. NYTimes maintains several Twitter accounts (e.g., *@nytimes, @nytpolitics, @nytopinion*), from which they regularly tweet the links to the stories published at `nytimes.com`. Using the Twitter streaming API [7], we collected the tweets made by the NYTimes accounts, and all retweets of these tweets. We also gathered the replies posted by the Twitter users who follow any of these NYTimes accounts. In total, we collected 1,026,116 posts during March 1, 2016 to April 30, 2016, and extracted links to 11,629 unique NYTimes stories. From this data, we computed the sequence of tweets (and retweets) mentioning each news story during this 2 month period.

# 3. UNDERSTANDING THE RECENCY-RELEVANCY TRADE-OFF

In this section, we first introduce the terminology used, then discuss the existing strategies for recommending news stories, and finally motivate the need for a new strategy.

## 3.1 Lifecycle of a news story: Terminology

Every news story in a media site goes through different phases in its **popularity lifecycle**, where the **popularity** of a story is usually based on some crowdsourced measure of readers' interest in that story. For instance, the popularity

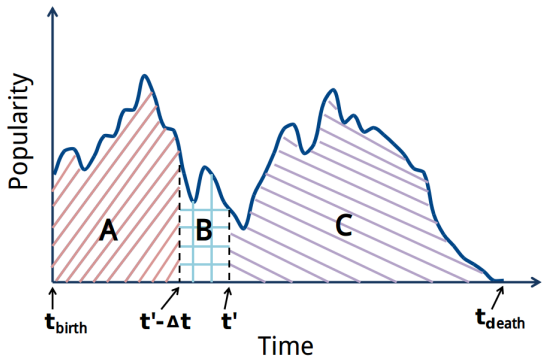



**Figure 1: Popularity lifecycle of a news story, where popularity can be measured as the number of views (or likes or shares) per unit time.**

of a story at time $t$ can be measured as the number of views (or likes or shares) the story gets in a unit time interval around $t$. Figure 1 shows the lifecycle of an example story $s$. $s$ appears in the media site at time $t_{birth}$, then receives different amounts of popularity at different time instants. Finally, its lifecycle gets over at time $t_{death}$, after which it does not get any more views. Thus, the **lifetime** of a story is the interval between the time instant when the story first appeared in the website and the instant when its lifecycle is over. For example, lifetime of $s$ is $|t_{death} - t_{birth}|$.

The **lifetime-impact** of a news story is measured by the number of views (or likes or shares) that the story gets during its entire lifetime. For example, the lifetime-impact of the story $s$ is the *area under the popularity curve* (i.e., the total area of the regions $A$, $B$ and $C$) in Figure 1.

Now, assume that a set of news stories are to be recommended at a particular time instant $t'$. The candidate set of stories comprises of all the news stories published before time $t'$, out of which different recommendation strategies would recommend different set of stories. The **age** of a recommended story is the difference between the time the story is published, and the time it is recommended. So, if the story $s$ is recommended at time $t'$, then the age of $s$ at the time of this recommendation would be $|t' - t_{birth}|$.

## 3.2 Desired properties of recommended news

In today's media sites, news stories appear at different points in time, and at the time of recommendation, a recommendation system picks $K$ stories from all stories published till that point in time.

Intuitively, it is desirable that the stories recommended by a recommendation strategy have two properties –

**(1) Recency:** The stories should be *recent*, i.e., their publish times should be close to the time of recommendation. In other words, the age of the recommended stories should not be high. To measure how a particular recommendation strategy performs in terms of recency, we can consider the average age of all stories recommended by this strategy.

**(2) Relevancy:** The recommended stories should also be *relevant*. Relevancy has two interpretations depending on whether the recommendation being *personalized* or *non-personalized*. For personalized recommendation, stories being relevant mean that the stories should match the interests of the user receiving the recommendation. For non-personalized recommendation, which we consider in this work, relevancy can be quantified by the lifetime-impact of the recommended stories. To measure how a particular recommendation strategy performs in terms of relevancy, we can compute average lifetime-impact of all stories recommended by this strategy.

| Recommendation Strategy | Average Age | Average Lifetime-Impact |
|---|---|---|
| 10 Latest Stories | 4.15 Hours | 1,684 Views |
| 10 Highest Lifetime-Impact Stories | 4.09 Days | 5,734 Views |

**Table 1: Comparing the performances of recommending latest stories and highest lifetime-impact stories.**

Ideally, recommended stories should simultaneously have high recency and high relevancy. But, as we show in the rest of the section, it is very hard to jointly optimize for recency and relevancy, when selecting news stories. In practice, we observe that when existing recommendation strategies perform better in one aspect, they tend to perform poorly on the other – we refer to this observation as the ***recency-relevancy trade-off***.

## 3.3 Recency-Relevancy trade-offs in existing recommendation strategies

We now describe some broad non-personalized recommendation strategies presently deployed in news media sites. While describing the strategies, for now, we assume the existence of an *'oracle'*, which knows the past as well as the future popularity of every news story published in the site. That is, at time $t'$, the oracle knows exactly how many views (or likes or shares) a story has received till $t'$, and how many it will receive after $t'$, throughout its entire lifetime.

### *3.3.1 Optimizing for recency OR relevancy*

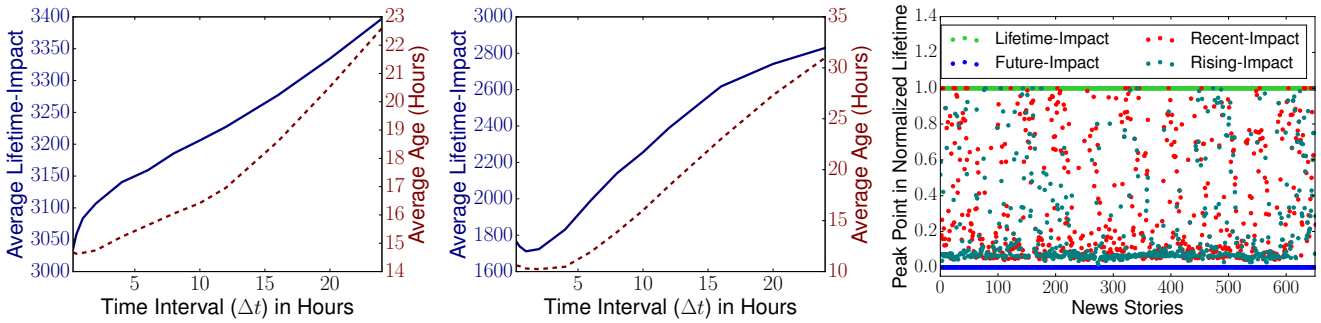
We start by describing some simple strategies that attempt to optimize for either recency or relevancy.

**Latest Stories:** In this strategy, the site simply recommends the most recent stories. All stories available at time $t'$ are ranked based on $|t' - t_{birth}|$, and then the $K$ latest stories are recommended.

**Highest Lifetime-Impact Stories:** Another strategy would be to recommend stories based on the lifetime-impact of the stories, i.e., based on the total number of views (or likes / shares) a story would receive during its entire lifetime (which we assume is known by the oracle). With respect to Figure 1, this strategy would rank all stories based on the total area under their popularity curves during the interval $[t_{birth}, t_{death}]$ (i.e., the combined area of regions $A$, $B$, and $C$), and then recommend the top $K$ stories.

Clearly, the two strategies described above are the two extremes. The strategy of recommending latest stories does not take into account the lifetime-impact of the stories, and hence might end up recommending stories which never become much popular. Whereas, recommending the highest lifetime-impact stories does not consider the recency of the stories, resulting in often recommending older stories at the end of their lifecycles.

Table 1 compares between the top 10 news stories recommended by the two extreme strategies on the Yahoo! News dataset described in Section 2. While the 10 Latest Stories have small average age (only 4.15 hours), their average lifetime-impact is also relatively low (1,684 views). On the other hand, the 10 Highest Lifetime-Impact Sto-




**(a)** **(b)** **(c)**

**Figure 2: Comparing the performances of recommending 10 stories with (a) highest recent-impact, and (b) highest rising-impact over different time intervals $\Delta t$. (c) The points in the normalized lifetimes where different news stories have highest values of lifetime-impact, future-impact, recent-impact and rising-impact. The recent-impact and rising-impact of the stories are computed over 1 hour (i.e., $\Delta t$ = 1 hour).**

ries have much higher lifetime-impact (5,734 views) but are much older (4.09 days).

### 3.3.2 *Trading between recency AND relevancy*

Between the two extreme strategies described above, there are other strategies that attempt to balance both recency and lifetime-impact, by looking at the popularity of news stories around the time of recommendation $t'$. We describe two such strategies next.

**Highest Recent-Impact Stories:** This strategy attempts to identify the stories that have the highest popularity (e.g., most viewed, most liked, or most shared) over a certain duration of time $\Delta t$ immediately before the recommendation instant $t'$. With respect to Figure 1, the stories will be ranked based on the area under their popularity curves during the interval $[t' - \Delta t, t']$ (i.e., the region $B$), and the top $K$ stories will be recommended. The choice of the interval $\Delta t$ varies widely in existing recommendation systems, ranging from last few minutes to few hours, last 1 day, or even last 1 month. Examples of this strategy include various recommendations available on the NYTimes site, e.g., most viewed over the last day, most shared over the last week, and so on [8].

Note that the choice of $\Delta t$ can have large implications on the type of stories being recommended. If $\Delta t$ is large, the recommended stories are older and the freshness of the stories are lost. Whereas, if $\Delta t$ is too small, it is not clear whether popularity over $\Delta t$ for a story is a good indicator of its lifetime-impact.

To bring out the implications of considering different values of $\Delta t$, Figure 2(a) compares sets of top 10 stories with highest recent-impact, considering various values of $\Delta t$ on the Yahoo! News data. As $\Delta t$ is increased from 15 minutes to 24 hours, the average age of the recommended stories increases, i.e., the stories gradually become less recent. But the average lifetime-impact of the stories increases, i.e., more relevant stories are recommended.

**Trending, or Highest Rising-Impact Stories:** Yet another recommendation strategy is based on how the popularity is changing over a certain duration of time $\Delta t$ immediately before the recommendation instant $t'$. This strategy is about picking the $K$ stories having the highest derivative (over time) of the popularity, computed over the duration $\Delta t$. In other words, the stories with highest rise in popularity during the last $\Delta t$ interval are recommended. Examples of this strategy include *Twitter Trending Topics* [9, 10].

But even here, the choice of $\Delta t$ is crucial in determining the type of stories being recommended. When $\Delta t$ is large, this strategy is similar to recommending highest recent-impact stories. Whereas, if $\Delta t$ is small, the recommendation strategy tends to pick *flash in the pan* stories which have high instantaneous peaks in popularity, and ignores stories gaining popularity more consistently. Figure 2(b) shows the average age and lifetime-impact for top 10 stories with highest rising-impact computed over different $\Delta t$ on the Yahoo! News data. Similar to the case with recent-impact in Figure 2(a), even here we observe the trend of increase in the age as well as the lifetime-impact of recommended stories with increase in $\Delta t$.

**Takeaway:** We see that different recommendation strategies attempt to balance between recency and relevancy in different ways. At a high level, the recency-relevancy tradeoff always exists, and increasing one usually leads to a fall in the other. Most existing implementations of these recommendation strategies use somewhat arbitrary parameters like $\Delta t$, such as 15 minutes for Twitter trending topics, 1 day for NYTimes most viewed stories over the last day, and so on. But it is not clear which strategy yields the *'best'* result, as the objective of these recommendations are not explicitly stated. In this work, we argue that we can adopt a new recommendation strategy which would help to get better recency as well as better relevancy. We present this strategy in the next section.

## 4. HIGHEST FUTURE-IMPACT RECOMMENDATIONS: A NEW STRATEGY

In this work, we propose a new recommendation strategy which will select stories based on their **future-impact**, i.e., *how much user attention (number of views or shares) each story is likely to receive in the future.* With respect to Figure 1, this strategy will rank all the stories available at time $t'$, based on the area under their popularity curves beyond time $t'$ (i.e., the region $C$), and choose the top $K$ stories to recommend. Note that, for now, we assume the presence of an *oracle* which has knowledge of the future. We will relax this assumption later, when we propose a practical implementation of the recommendation strategy in the next section.

| Recommendation Strategy | Yahoo! News | | | CLEF NewsREEL | | | NYTimes | | |
|---|---|---|---|---|---|---|---|---|---|
| | Average Age (Hours) | Average Lifetime-Impact | Average Future-Impact | Average Age (Hours) | Average Lifetime-Impact | Average Future-Impact | Average Age (Hours) | Average Lifetime-Impact | Average Future-Impact |
| Latest | **4.72** | 1729.51 | 1346.15 | **1.26** | 3.47 | 3.11 | **2.28** | 26.99 | 14.64 |
| Highest Lifetime-Impact | 71.79 | **5211.95** | 461.05 | 341.4 | **211.64** | 44.84 | 141.77 | **269.72** | 15.61 |
| Highest Rising-impact | 10.96 | 1832.06 | 875.38 | 238.76 | 47.39 | 33.56 | 2.88 | 63.96 | 20.37 |
| Highest Recent-impact | 22.93 | 3425.98 | 539.92 | 255.42 | 77.46 | 27.06 | 18.39 | 119.16 | 21.55 |
| Highest Future-impact | 7.71 | 2841.06 | **1835.76** | 225.36 | 112.84 | **92.41** | 17.69 | 117.72 | **30.91** |

**Table 2: Comparing the performances of recommending 10 latest, highest lifetime-impact, highest rising-impact, highest recent-impact, and highest future-impact stories. Best values for each metric are in bold blue. The $\Delta t$ duration for the highest rising-impact and highest recent-impact stories are taken to be 15 minutes and 24 hours respectively.**

## 4.1 Why recommend stories based on Highest Future-Impact?

We now describe two main motivations for recommending news stories having the highest future-impact.

**(1) The normative argument:** The effectiveness of recommending news stories based on their future-impact can be argued *normatively* using several communication theories, which consider the social aspects of reading news. Using the seminal work of Habermas *et al.* [11] on the *'public sphere'*, Novendstern [3] argued that news is a part of people's *'public discourse'*, using which they can participate in community discussions. Therefore, a reader should read those stories which will be largely discussed in future, rather than the stories which have already been discussed in the past.

On one hand, applying the *'knowledge gap hypothesis'* [12], we can argue that if some readers read interesting news stories ahead of their peers, such differences in knowledge acquisition help maintain the knowledge gap between different segments of the society. However, on the other hand, using the advance knowledge, such readers can play the roles of *opinion leaders* [13], and initiate discussions in their communities around the news stories. Such spreading of ideas from mass media to opinion leaders, and from them to the wider society, forms the basis of the *two-step flow of communication model* [14]. Therefore, a recommendation strategy *should* recommend to its readers the stories which would enable such information flow.

**(2) Better recency-relevancy trade-off:** The second motivation for recommending stories with highest future-impact comes from the perspective of recency-relevancy trade-off. Unlike the existing strategies which optimizes for one at the cost of the other, the Highest Future-Impact strategy can optimize for *both* recency and relevancy of the recommended news stories.

The future-impact of every story declines over time, from the maximum at its birth[2] to zero at its death. Hence, story selection based on its future-impact effectively captures the trade-off between its age (recency) and its lifetime-impact (relevancy). The strategy would like to pick stories with *high* lifetime-impact and that too *early* in their lifetimes, which is different from existing strategies of selecting stories based on their lifetime-impact, which stays the same throughout a story's lifetime, or their recent-impact or rising-impact, that are *not* guaranteed to decrease over time.

To demonstrate this difference, we normalized the lifetime of every news story such that any time instant in its lifecycle would fall between 0 and 1. Then, we checked at what point in its lifecycle, the story has the highest value of recent-impact, rising-impact, and the future-impact. Figure 2(c) shows the highest points for different stories in the Yahoo! News dataset, where the $y$-axis shows the normalized lifetime of stories. We can see that the highest future-impacts for all stories are at time 0 (blue colored points at $y = 0$). Although the lifetime-impacts for all stories remain the same throughout the lifetime, it will be *fully known* only at time 1 (light green colored points at $y = 1$). Regarding recent-impact and rising-impact, different stories reach their highest values at different points of time during their life-cycle; often long after they are published, hence, the corresponding highest points are scattered throughout Figure 2(c).

[2]At birth, future-impact of a story equals its lifetime-impact.

## 4.2 Comparing Highest Future-Impact strategy with existing strategies

To compare different recommendation strategies mentioned earlier, we execute the strategies over the stories which first appeared during the initial 70% of our datasets (chronologically ordered), and received no views (or shares) during the last 10% of the data. Rest of the stories are not considered as the lifetimes of these stories may not be over; hence, it will not be possible to know the actual lifetime-impact and the future-impact values for them. We consider the lifetime of a story to be over when it does *not* receive any view (or share) during the rest of the datasets.

We execute different recommendations at every 15-minute intervals over the time duration covered by the initial 70% of the datasets, and pick the top 10 stories as recommended by different strategies. We then compute different performance metrics for the recommended stories, and the average value of these metrics are used for comparison.

The comparison is along three different metrics – (i) average age of the recommended stories (which captures recency), (ii) average lifetime-impact of the recommended stories (which captures relevancy), and (iii) average future-impact of the recommended stories.

Table 2 shows the average performance of recommending news stories according to different strategies over Yahoo! news, CLEF NewsREEL, and NYTimes datasets. Table 2 demonstrates the recency-relevancy trade-off. The strategy which achieves the maximum lifetime-impact (Highest Lifetime-Impact) suffers from high average age of the recommended stories, while the strategy which achieves lowest average age (Latest) has the lowest average lifetime-impact. Other strategies, like Highest Recent-Impact and Highest Rising-Impact, achieve some balance along these two metrics. However, the Highest Future-Impact strategy often achieves good performance with respect to both metrics. Additionally, the Highest Future-Impact strategy also gives stories which will get most attention in future.

**Takeaway:** The results indicate the benefits of the Highest Future-Impact strategy over the existing recommenda-

tion strategies. Additionally, maximizing future-impact has a clear incentive for media sites wanting to maximize advertising revenues. If they can recommend stories which bring in the highest number of views in the future, that in turn will maximize their revenue as well.

# 5. IMPLEMENTING HIGHEST FUTURE-IMPACT RECOMMENDATIONS

As stated earlier, we have assumed the existence of an omniscient oracle till now, which has the knowledge of the future-impact of every story. Next, we focus on actually implementing the Highest Future-Impact recommendation strategy.

To recommend the stories having highest future-impact, in practice, we would need to *estimate* the future-impact of a story. So, one simple strategy would be to *predict the future-impact* of all stories at a particular time instant, and recommend the stories which have the highest predicted future-impact. Section 5.1 presents this strategy in detail.

However, with any prediction task, there is an associated *uncertainty* regarding how accurate the predicted future-impacts are (Section 5.2). If the predicted values are too different from the actual future-impact values, then this strategy might either end up recommending stories which do not have much future-impact, or miss the stories which actually have high future-impact. Therefore, we present another strategy (detailed in Section 5.3), where we attempt to minimize the uncertainties in the predicted future-impact of the recommended stories.

## 5.1 Recommendations using future-impact predictions

When a news story is published at time $t_{birth}$, we only have the textual content and some meta-information of the story (e.g., its topical category, the event on which the story is reporting, the author of the story, and so on). As time progresses, we get the information on how the readers are interacting with the story. For example, we can divide the time starting from $t_{birth}$ in different fixed $t$-sized time intervals (e.g., $t$ can be 5, 15, 30 minutes or longer), and then compute its popularity (e.g., the number of views the story got) during these time intervals.

To predict the future-impact of a story, we first attempt to predict the lifetime-impact of the story. Then, the predicted future-impact can be computed as the predicted lifetime-impact, minus the number of views (or likes or shares) the story has received so far. Thus, for a given news story $s$, our task is to predict the lifetime-impact at time $\tau$ using the information available till time $\tau$.

This prediction task falls under the broad class of estimating the amount of user attention for different online contents. There have been attempts to predict the user attention for Youtube videos [15], Flickr images [16] and so on. However, there is one distinction which makes the prediction for news stories different than other types of contents. The lifetimes of news stories are much smaller compared to the lifetimes of other types of contents, and due to this very nature of news, it is desirable to accurately predict the lifetime-impact as early as possible from the publish time $t_{birth}$, and with only a limited amount of data.

Due to this constraint, several past works on online news [17–19] have attempted to predict user attention classes (e.g., whether a story is going to be viral or not) instead of predicting the exact amount of user attention. However, in our context, coarse grained user attention classes will be insufficient to give us an estimate of the lifetime-impact of stories.

Additionally, due to the limitation of the Yahoo! News data we are using in this work, we could not extract any content or meta-information for the news stories. Hence, for the sake of generality, regardless of the dataset, we attempt to predict the lifetime-impact of the news stories at time $\tau$, only using the number of views (or shares in case of NYTimes data) that the stories received between their publish times and $\tau$.

Specifically, for a news story $s$, we first compute the feature vector $x_s$ of size $m$, where $m$ is the number of 15 minute intervals between $t_{birth}$ and $\tau$, and each feature in $x_s$ is the number of views (or shares) $s$ received during the corresponding interval. Then, we predict the lifetime-impact $y_s$ using this feature vector $x_s$ as input. We explore two methods to predict the lifetime-impact of news stories, which we describe next.

**Method 1: Ordinary Least Squares (OLS)**
In the first method, we predict $y_s$ assuming a linear model: $y_s = x_s^T\beta + \epsilon_s$, where $\beta$ is the vector of weights for different features including the intercept $\beta_0$, and $\epsilon_s$ is the random noise with zero mean and constant variance $\sigma^2$. $\beta$ is then estimated by minimizing the *sum of squared errors* for a set of $n$ stories (training data points), for which the lifetime-impact is known a priori [20]. Specifically, the *least squares estimate* of $\beta$ (denoted as $\hat{\beta}$) is measured as

$$\hat{\beta} = \arg\min_{\beta} \frac{1}{2n} \sum_{i=1}^{n} (y_i - x_i^T\beta)^2 \quad (1)$$

where $y_i$ and $x_i$ are the lifetime-impact and the feature vector for story $i$ respectively.

Once we get the estimated weight vector $\hat{\beta}$, then given the observed feature values $x_s$ for story $s$, the predicted lifetime-impact $\hat{y_s}$ is computed as

$$\hat{y_s} = x_s^T\hat{\beta} \quad (2)$$

where $\hat{y_s}$ is the *conditional mean* $E(y_s|x_s)$.

**Method 2: Gradient Tree Boosting (GTB)**
In the first method, we assume that $y_s$ can be expressed as a linear combination of features in $x_s$. However, if this assumption is not valid in the real data, then the linear model will fail to capture the reality and as a result, OLS will have lower accuracy in predicting $y_s$. In method 2, we use non-parametric regression model *Decision Trees*, which does not assume anything about the nature of the underlying relationship between $y_s$ and the features $x_s$ [21].

Although decision trees can work without having any underlying assumption of the data, Breiman [22] showed that decision trees are *unstable* in the sense that small perturbations in the training set may result in large changes in the constructed predictor. To improve the accuracy, Breiman argued for using an ensemble of multiple decision trees (e.g., 10, 100, or 500 such trees) instead of using only one predictor, and then combining their individual predictions. Gradient Tree Boosting (GTB) [23] is one of the ways to do exactly that.

GTB starts with *short* decision trees (also called *weak learners*) to predict $y_s$, and gradually adds larger trees using a gradient descent like procedure. In each addition step, a tree is added to the model which minimizes a particular loss function computed over the training samples. The final predicted lifetime-impact $\hat{y_s}$ is computed as the weighted sum

| Recommendation Strategy | Yahoo! News | | | CLEF NewsREEL | | | NYTimes | | |
|---|---|---|---|---|---|---|---|---|---|
| | Average Age | Average Lifetime-Impact | Average Future-Impact | Average Age | Average Lifetime-Impact | Average Future-Impact | Average Age | Average Lifetime-Impact | Average Future-Impact |
| Latest | 4.89 | 1820.66 | 1375.5 | 2.29 | 5.64 | 5.21 | 1.47 | 24.2 | 11.48 |
| Highest Lifetime-Impact | 96.7 | 5583.02 | 456.69 | 389.14 | 191.69 | 32.4 | 168.43 | 221.4 | 13.29 |
| Highest Rising-impact | 12.33 | 1949.11 | 926.67 | 238.23 | 57.45 | 41.59 | 3.22 | 76.02 | 14.27 |
| Highest Recent-impact | 24.01 | 3528.75 | 576.74 | 275.82 | 85.63 | 37.62 | 19.61 | 132.64 | 17.87 |
| Highest Future-impact | 8.05 | 2968.56 | 1912.01 | 231.53 | 119.73 | 87.46 | 16.74 | 151.59 | 27.6 |
| Prediction using OLS | 9.09 | **2781.93** | **1621.8** | **236.33** | **124.32** | **79.61** | **8.88** | **145.29** | **22.62** |
| Prediction using GTB | **8.01** | 2693.55 | 1619.37 | 247.77 | 101.53 | 71.48 | 15.37 | 143.16 | 19.91 |

**Table 3: Comparing the performances of two future-impact prediction methods with other recommendation strategies.**

of the predictions from the sequence of trees being added. The benefit of GTB is that it can work with any differentiable loss function, e.g., least squares, least absolute deviation, etc. In this work, we particularly use least absolute deviation as the loss function for GTB.

**Predicted future-impact:** Finally for both methods, once we get the predicted lifetime-impact of $s$, we compute the predicted future-impact $f_\tau(s)$ of $s$ at time $\tau$ as

$$f_\tau(s) = \hat{y_s} - \sum_{t=t_{birth}}^{\tau} popularity_t(s) \quad (3)$$

where $popularity_t(s)$ is the number of views (or shares) obtained by $s$ at time $t$.

**Comparing different methods for prediction**

We now compare the performance of recommending stories using the predictions made by the above two methods. We first predict the future-impact of stories using both methods. Then, all stories are ranked based on the predicted future-impact of the stories and top $K$ stories are recommended.

As explained in Section 4.2, we only consider the stories appearing in the initial 70% of the datasets. Among them, we use the stories in first 40% of the datasets as training, and the stories appearing in the next 30% as the test data to compare the performances. We execute different recommendations at every 15 minute intervals over the test data, and compute different performance metrics on the 10 stories recommended by different strategies.

We compare the performances along the three metrics introduced earlier – (i) average age, (ii) average lifetime-impact, and (iii) average future-impact of the recommended stories. Table 3 shows the average performance of recommending based on the future-impact values predicted by the two methods and all other strategies mentioned in the earlier section. We can see from Table 3 that the future-impact prediction using both methods work well, yielding results comparable to the highest future-impact stories. Between the two, prediction using OLS outperforms the prediction using GTB by achieving performances closer to the strategy of recommending highest future-impact stories, only except the recency of the recommended Yahoo! news stories.

## 5.2 Uncertainty in future-impact predictions

Figure 3 shows how the actual future-impacts and the values predicted by OLS change with time, for three example Yahoo! news stories. For some stories (e.g., Figure 3(a)), the predicted future-impact quickly converges to the actual future-impact. For other stories, the prediction comes close to actual future-impact much later in the lifetime (Figure 3(b)), or sometimes they never match at all (Figure 3(c)).

One way to control this risk of mis-prediction is to recommend stories where the uncertainty in predicted future-impact is less. We now describe how to estimate the uncertainty in the prediction.

**Uncertainty in prediction using OLS**

Recall that in OLS the lifetime-impact $y_s$ is assumed to be of the form $y_s = x_s^T\beta + \epsilon_s$. Whereas, the predicted lifetime-impact is $\hat{y_s} = x_s^T\hat{\beta}$. Hence, the prediction error is

$$y_s - \hat{y_s} = [x_s^T\beta - x_s^T\hat{\beta}] + \epsilon_s \quad (4)$$

Due to the introduction of random noise $\epsilon_s$ and the difference in the estimated $\hat{\beta}$ and actual $\beta$, it may *not* be enough to only use the conditional mean $E(Y_s|x_s)$ value as the point estimate of the lifetime-impact. Rather, it might be more useful to specify a range or an interval which has a high probability of containing the actual lifetime-impact value, i.e., to specify the ***Prediction Interval***. The 100 $(1-\alpha)$% prediction interval for OLS [20] is given as

$$x_s^T\hat{\beta} \pm t_{n-m}^{(\alpha/2)} Var\ (y_s - \hat{y_s}) = x_s^T\hat{\beta} \pm t_{n-m}^{(\alpha/2)}\hat{\sigma}\sqrt{1 + x_s^T(X^TX)^{-1}x_s} \quad (5)$$

where $X$ is the matrix (of dimension $n \times m$) containing the feature vectors for $n$ stories used for training; $\hat{\sigma}^2$ is the estimated variance of the random noise $\epsilon_s$, and $t_{n-m}^{(\alpha/2)}$ is the value of *Student's t-distribution* [24] for specific values of $\alpha$, $n$ and $m$. In this work, we have used the width of the 90% prediction interval (i.e., $\alpha = 0.1$) as the measure of the uncertainty in prediction.

**Uncertainty in prediction using GTB**

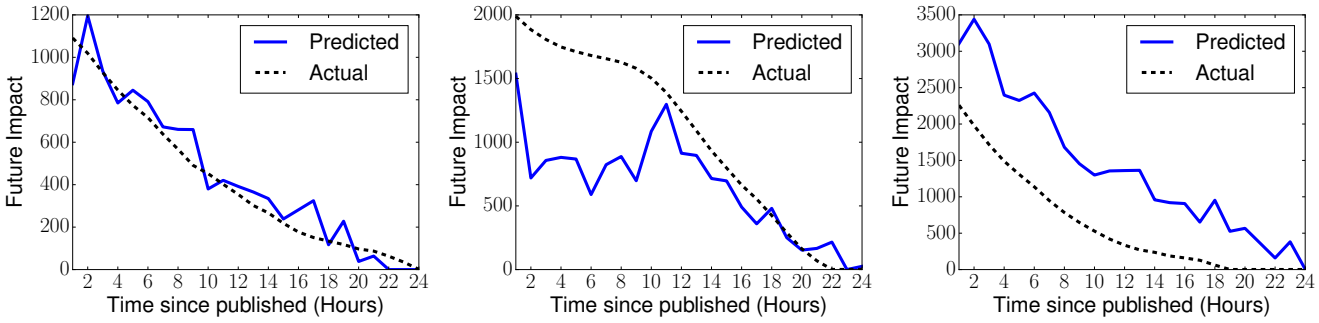
As mentioned earlier, one of the benefits of using GTB is that we can apply any loss function for building the model. In order to get the prediction interval for GTB, we use the loss function for **Quantile Regression** [25]. In Linear Regression, the predicted lifetime-impact $\hat{y_s}$ is the conditional mean of the response variable $y_s$ for given values of the input $x_s$. In Quantile Regression, instead of estimating the conditional mean, we can estimate the *conditional quantile* $Q_q(y_s|x_s)$ of the response variable $y_s$, for any quantile $q$ [25].

Given $n$ training data points, in GTB, the quantile loss function for quantile q can be expressed as

$$\sum_{i:y_i<\hat{y_i}}^{n} (1-q)(y_i - \hat{y_i}) + \sum_{i:y_i\geq\hat{y_i}}^{n} q(y_i - \hat{y_i}) \quad (6)$$

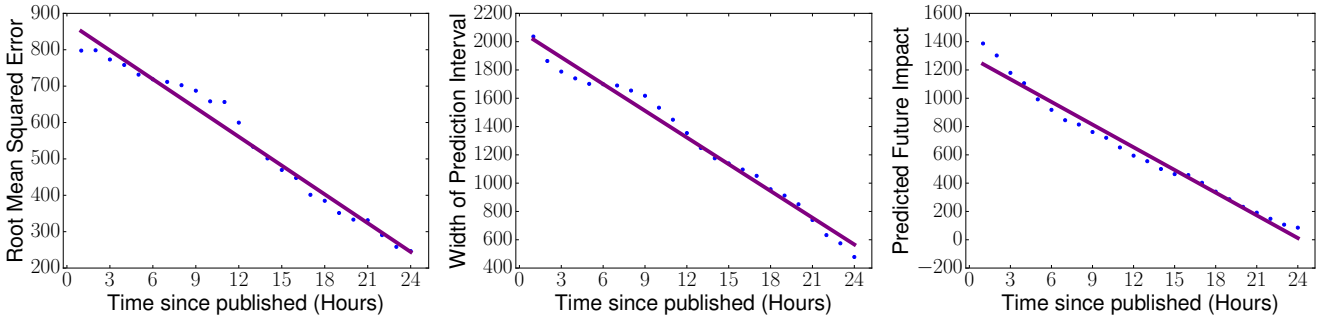
where $y_i$ and $\hat{y_i}$ are the actual and predicted lifetime-impact for news story $i$. While sequentially building the model, we can configure GTB to minimize this loss function.

The 100 $(1-\alpha)$% prediction interval for GTB can be computed by taking the predicted values by two GTB models built using the quantile loss functions for quantiles $\alpha$ and $1-\alpha$ respectively. If we assume the predicted values to be $\hat{y_{s_1}}$ and $\hat{y_{s_2}}$ respectively, then the 100 $(1-\alpha)$% prediction interval is computed as $|\hat{y_{s_1}} - \hat{y_{s_2}}|$. Similar to OLS, we use the width of the 90% prediction interval (i.e., $\alpha = 0.1$) as the measure of uncertainty in the prediction by GTB.




(a) Good match early in lifetime (b) Good match late in lifetime (c) No match

**Figure 3: Comparing predicted and actual values for future-impact for different stories at different points in time.**




(a) (b) (c)

**Figure 4: Change with time in (a) Root Mean Squared Error (RMSE), (b) Width of the Prediction Interval, and (c) Predicted future-impact. The (blue) dots in the figures represent the values and the (purple) lines show the trends.**

## 5.3 Accounting for prediction uncertainty in recommendations

As discussed earlier, to get the highest future-impact from the recommended stories, we need to pick those stories for which the predicted future-impact is close to the actual future-impact. Figure 4(a) shows how the difference between the actual future-impact and the future-impact predicted by OLS changes over time, when the values are averaged over a large number of stories from the Yahoo! news dataset. Here, the difference is measured in terms of Root Mean Squared Error (RMSE), defined as $RMSE = \sqrt{\frac{\sum_{i=1}^{w}(\hat{y}_i - y_i)^2}{w}}$, where $w$ is the number of stories used for testing.

As time progresses, more information becomes available on how users are interacting with a story. Thus, the prediction model becomes richer and hence, the predicted future-impact becomes closer to the actual future-impact of the story. In other words, with increasing time, the predicted future-impact converges to the actual future-impact, and the uncertainty in the prediction (width of the prediction interval) becomes smaller. This is shown in Figure 4(b).

However, we can see in Figure 4(c) that the predicted future-impact for a story also decreases with time. So, if a certain story is recommended late in its lifetime (rather than early), the uncertainty in prediction of future-impact will be lower; however, at the same time, the predicted future-impact value itself will be smaller. Therefore, we need a systemic approach to balance between the predicted future-impact and the uncertainty in prediction.

We propose the following optimization framework to recommend stories at time $\tau$, which minimizes the sum of the uncertainties in the future-impact predictions, by utilizing a small leeway in maximizing the cumulative predicted future-impact values of the recommended stories.

$$minimize \sum_{i=0}^{N_c} u_\tau\ (s_i) \cdot z_i \tag{7}$$

where $u_\tau\ (s_i)$ is the uncertainty associated with predicting the future-impact value $f_\tau\ (s_i)$ for story $s_i$ at time $\tau$, $N_c$ is the number of candidate stories, and $\{z_i\}$ are the indicator random variables (if $z_i = 1$, then $s_i$ is recommended).
subject to the constraints

$$z_i \in \{0, 1\} \tag{8}$$

$$\sum z_i = K \tag{9}$$

$$\sum f_\tau\ (s_i) \cdot z_i \geq \gamma \cdot {f_\tau}^* \tag{10}$$

where $K$ is the number of stories to be recommended, ${f_\tau}^*$ is the maximum possible predicted future-impact values for $K$ news stories at time $\tau$, and $\gamma \in [0, 1]$ controls the maximum amount of leeway allowed in the predicted future-impacts of the recommended news stories. In this work, we take $\gamma = 0.99$ in our experiments.

## 5.4 Comparing the performances of prediction with and without uncertainty

We executed both recommendation strategies – one which uses only the predicted future-impact, and the other which takes into account the uncertainty in prediction – using the stories for training and testing as described in Section 5.1. To solve the minimization problem detailed in Equation 7, we use the state-of-the-art linear programming solver Gurobi [26].

In our comparison over three datasets, we observe that the average age of stories recommended by prediction with un-

| Dataset | Prediction Method | Change in Future-Impact | | |
|---|---|---|---|---|
| | | Increase | No change | Decrease |
| Yahoo! News | OLS | 25.22 | 22.08 | 52.7 |
| | GTB | 27.92 | 18.12 | 53.96 |
| CLEF NewsREEL | OLS | 28.92 | 23.45 | 47.63 |
| | GTB | 26.38 | 30.11 | 43.51 |
| NYTimes | OLS | 31.6 | 28.8 | 39.6 |
| | GTB | 23.37 | 45.24 | 32.38 |

**Table 4: Effect of considering prediction uncertainty: % of the recommendation instants where actual future-impact (a) increased, (b) did not change, and (c) decreased compared to prediction without uncertainty.**

certainty increases marginally. Because, choosing an older article instead of a new article often makes the overall uncertainties less. However, on the other hand, average lifetime-impact of the recommended stories are also higher in case of prediction with uncertainty.

Regarding the future-impact, even though the prediction with uncertainty approach can pick stories having lower predicted future-impact to minimize the prediction uncertainty, Table 4 shows that in a large percentage of the recommendation instants, the actual future-impact values were higher in the recommended stories. Additionally, when the stories having highest predicted future-impact have lower prediction uncertainties, both approaches tend to recommend same set of stories, thereby ensuring no loss of performance.

Digging deeper, we found that considering prediction uncertainties yield better results when the arrival rates of new stories as well as the traffic pattern deviates significantly from the standard patterns (for example, when some major event breaks out). As tools have been proposed (e.g., [27]) to detect such breakout of events, the news recommendation designers can switch between prediction with or without uncertainty to get optimum result in such scenarios.

**Takeaway:** The basic idea behind deploying the Highest Future-Impact strategy in practice is to *predict* the future-impact of a story at time $T$, based on the observed impact of the story till time $T$. However, the key difficulty lies in estimating and accounting for the *uncertainty in the future-impact predictions*, which can be large soon after the story's publication, but decreases over time. The evaluation results show that the proposed approach for predicting the future-impact of recommended stories, as well as the proposed framework of minimizing the cumulative uncertainties, successfully tackle the challenges and achieve good performance trade-offs between recency and relevancy.

# 6. RELATED WORK

**Personalized vs. non-personalized news recommendations:** A lot of prior works have focused on developing *personalized* news recommendation systems, which recommend news stories tailored to individual users. For example, Liu *et al.* [28] developed a Bayesian model to predict individual user's interests from her past activities, and the news trend reflected from the activities of a group of users, and then recommend stories according to the interests. Li *et al.* [29] designed a scalable personalized news recommender system by using a two-level representation, containing the topics relevant to user's preference at one level, and the news articles on these topics at the second level. Agarwal *et al.* [30] proposed *click shaping* to jointly optimize the number of clicks and post-click downstream utilities for recommending news stories. Maksai *et al.* [31] proposed metrics to evaluate the performance of such systems. However, except the efforts like *Most Emailed* stories [8] or *Trending Topics* [9] deployed in media sites today, there are not many research works to develop *non-personalized* news recommendation systems. In this work, we attempt to fill this vacuum by presenting a systematic approach to recommend news stories in the non-personalized scenario.

**Predicting popularity of online contents:** Prior works have attempted to predict the popularity of Youtube videos [15], Flickr images [16], or future citation count of research papers [32, 33]. Similarly, efforts have been made to predict popularity classes [17, 19], understand different popularity dynamics [34, 35], and whether they lead to the emergence of self-fulfilling prophecies [36], or social influence biases [37]. Complementary to the above works, in this paper, our focus is on predicting the user attention different stories are going to receive in future, and utilize them for recommending news stories.

**Recency vs Relevancy debate:** Present approaches on designing content recommendation systems are putting increasing emphasis on the recency and realtimeness of content. For example, Liang *et al.* [38] proposed a time-aware content recommendation system, while Watanabe *et al.* [27] proposed a framework to detect breaking news, and trending events from Twitter. This focus on recency also leads to a growing concern over the long-term importance of the recommended contents, and many users view such content as potentially waste of time information [39]. Although this debate on recency versus relevancy is going on for some time, to our knowledge, we are the first to propose a recommendation strategy which can simultaneously optimize for recency as well as the relevancy of the recommended news stories.

# 7. CONCLUSION

Many online news media sites are increasingly deploying automated recommender systems to constantly update their front-page stories based on crowd-driven measures of popularity of the stories. In this work, we focus on the fundamental tension faced by any recommendation strategy between choosing the most recent stories versus the most relevant or impactful stories. We conduct a systematic analysis of the recency-relevancy trade-offs achieved by the currently deployed recommendation strategies over three real world news datasets. After inferring the reasons for their poor performance, we propose a simple yet previously overlooked strategy of recommending stories based on their *future-impact*. We propose and evaluate a practical implementation of the future-impact based recommendation strategy, tackling the tricky challenge of estimating and accounting for uncertainties in predicting future-impact of stories.

Throughout this paper, we have relied only on the user-news engagements, and did not utilize the news content for predicting future-impact. Our future work lies in investigating whether considering content-based features leads to increase in the recency of the recommended news stories.

**Acknowledgments:** We thank the reviewers whose suggestions helped to improve the paper. This research was partially supported by a grant from the Information Technology Research Academy (ITRA), DeITY, Government of India (Ref. No.: ITRA/15(58)/Mobile/DISARM/05). A. Chakraborty was supported by Google India PhD Fellowship and Prime Minister's Fellowship for Doctoral Research.

# 8. REFERENCES


[1] Pew Research Center. Amid Criticism, Support for Media's 'Watchdog' Role Stands Out. goo.gl/gbjP0Q.

[2] Abhijnan Chakraborty, Saptasrshi Ghosh, Niloy Ganguly, and Krishna P. Gummadi. Can trending news stories create coverage bias? on the impact of high content churn in online news media. In *Computation and Journalism Symposium*, 2015.

[3] Max Novendstern. Why do we read the news? *Harvard Political Review*, 2011.

[4] Yahoo Webscope datasets. `webscope.sandbox.yahoo.com`.

[5] Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *ACM WSDM*, 2011.

[6] Frank Hopfgartner, Torben Brodt, Jonas Seiler, Benjamin Kille, Andreas Lommatzsch, Martha Larson, Roberto Turrin, and András Serény. Benchmarking news recommendations: The clef newsreel use case. In *ACM SIGIR Forum*, volume 49, 2016.

[7] Twitter Streaming API. `dev.twitter.com/streaming/overview`.

[8] Abhijnan Chakraborty, Saptarshi Ghosh, Niloy Ganguly, and Krishna P Gummadi. Dissemination biases of social media channels: On the topical coverage of socially shared news. In *ICWSM*, 2016.

[9] Twitter. To Trend or Not to Trend. blog.twitter.com/2010/to-trend-or-not-to-trend, 2010.

[10] Michael Mathioudakis and Nick Koudas. Twittermonitor: Trend detection over the twitter stream. In *ACM SIGMOD*, 2010.

[11] Jürgen Habermas, Sara Lennox, and Frank Lennox. The public sphere: An encyclopedia article. *New German Critique*, (3), 1974.

[12] Phillip J Tichenor, George A Donohue, and Clarice N Olien. Mass media flow and differential growth in knowledge. *Public opinion quarterly*, 34(2), 1970.

[13] Marsha L Richins and Teri Root-Shaffer. The role of evolvement and opinion leadership in consumer word-of-mouth: An implicit model made explicit. *NA-Advances in Consumer Research*, 1988.

[14] Elihu Katz. The two-step flow of communication: An up-to-date report on an hypothesis. *Public opinion quarterly*, 21(1), 1957.

[15] Flavio Figueiredo, Jussara M. Almeida, Marcos André Gonçalves, and Fabrício Benevenuto. On the dynamics of social media popularity: A youtube case study. *ACM Transactions on Internet Technology*, 2014.

[16] Philip J McParlane, Yashar Moshfeghi, and Joemon M Jose. Nobody comes here anymore, it's too crowded; predicting image popularity on flickr. In *ACM ICMR*, 2014.

[17] Abhijnan Chakraborty, Bhargavi Paranjape, Sourya Kakarla, and Niloy Ganguly. Stop clickbait: Detecting and preventing clickbaits in online news media. In *IEEE/ACM ASONAM*, 2016.

[18] Roja Bandari, Sitaram Asur, and Bernardo A Huberman. The pulse of news in social media: Forecasting popularity. In *AAAI ICWSM*, 2012.

[19] Julio Reis, Fabricio Benevenuto, Pedro Vaz de Melo, Raquel Prates, Haewoon Kwak, and Jisun An. Breaking the news: First impressions matter on online news. In *ICWSM*, 2015.

[20] Julian Faraway. Practical regression and anova using r.

[21] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[22] Leo Breiman. Bias, variance, and arcing classifiers. *STATISTICS*, 1996.

[23] Jerome Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 2001.

[24] Charles W Dunnett and Milton Sobel. A bivariate generalization of student's t-distribution, with tables for certain special cases. *Biometrika*, 1954.

[25] Roger Koenker. *Quantile regression*. 2005.

[26] Gurobi Optimization et al. Gurobi optimizer reference manual. *www.gurobi.com*, 2, 2012.

[27] Kazufumi Watanabe, Masanao Ochi, Makoto Okabe, and Rikio Onai. Jasmine: A real-time local-event detection system based on geolocation information propagated to microblogs. In *ACM CIKM*, 2011.

[28] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. Personalized news recommendation based on click behavior. In *ACM IUI*, 2010.

[29] Lei Li, Dingding Wang, Tao Li, Daniel Knox, and Balaji Padmanabhan. Scene: a scalable two-stage personalized news recommendation system. In *ACM SIGIR*, 2011.

[30] Deepak Agarwal, Bee-Chung Chen, Pradheep Elango, and Xuanhui Wang. Click shaping to optimize multiple objectives. In *ACM KDD*, 2011.

[31] Andrii Maksai, Florent Garcin, and Boi Faltings. Predicting online performance of news recommender systems through richer evaluation metrics. In *RecSys*, 2015.

[32] Rui Yan, Jie Tang, Xiaobing Liu, Dongdong Shan, and Xiaoming Li. Citation count prediction: learning to estimate future citations for literature. In *CIKM*, 2011.

[33] Xiao Yu, Quanquan Gu, Mianwei Zhou, and Jiawei Han. Citation prediction in heterogeneous bibliographic networks. In *SIAM ICDM*, 2012.

[34] Janette Lehmann, Bruno Gonçalves, José J Ramasco, and Ciro Cattuto. Dynamical classes of collective attention in twitter. In *WWW*, 2012.

[35] Riley Crane and Didier Sornette. Robust dynamic classes revealed by measuring the response function of a social system. *PNAS*, 2008.

[36] Matthew J Salganik and Duncan J Watts. Leading the herd astray: An experimental study of self-fulfilling prophecies in an artificial cultural market. *Social psychology quarterly*, 2008.

[37] Lev Muchnik, Sinan Aral, and Sean J Taylor. Social influence bias: A randomized experiment. *Science*, 2013.

[38] Huizhi Liang, Yue Xu, Dian Tjondronegoro, and Peter Christen. Time-aware topic recommendation based on micro-blogs. In *ACM CIKM*, 2012.

[39] Stop Overdosing on Celebrity Gossip, The News, and Low Quality Information. jamesclear.com/brain-food.