Accounting for the Impact of Media Coverage on Polling

Arun Jose
joseverres42@gmail.com

Abstract

This paper examines the feedback cycle of news ratings and electoral polling, and offers an algorithmic news algorithm to patch the problem. The cycle hinges on overexposure of a candidate to familiarize their name in otherwise apathetic voters, and therefore, the algorithm weighs down exposure on a logarithmic scale to only pass increasingly important news as coverage of a candidate inflates.

This problem is a symptom of a deeper issue, and the solution proposes to patch it for the present, as well as offer insight into the machinations of the issue, and therefore aid its understanding.

Contents

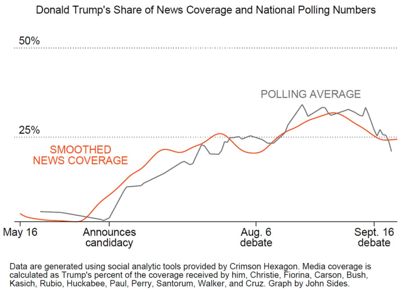
1 Introduction

Media is a high-level system built on high-level players, and is thus highly susceptible to increasingly damaging flaws. The 2016 American elections portrayed a novel defect in the relationship between media and politics.

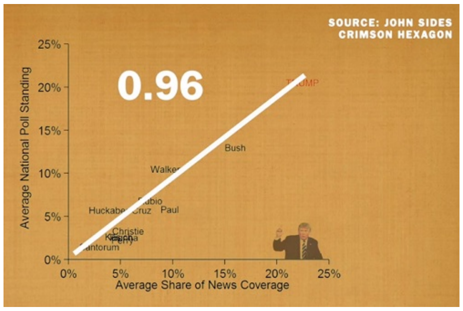
1.1 Feedback Loops

In the four months following the announcement of his candidacy in 2015, Donald Trump was the subject of over 2100 CNN reports alone. The basic premise is simple: he gets ratings, so he gets news. But there's a deeper factor at play: Trump's high poll numbers are a result of his outlier media coverage.

The following graph shows the correlation between Trump's share of coverage versus his poll numbers.



Data are generated using social analytic tools provided by Crimson Hexagon. Media coverage is calculated as Trump's percent of the coverage received by him, Christie, Fiorina, Carson, Bush, Kasich, Rubio, Huckabee, Paul, Perry, Santorum, Walker, and Cruz. Graph by John Sides.



[1]

The causation could be marked the other way, with polling driving news coverage, but the coverage usually precedes the polls.

The reason for this is familiarization: a person's name is read over and over again, until it's ingrained even - and especially - in the minds of those otherwise apathetic toward keeping up with political news. This moves them to give that name to pollsters, and thereby increasing news coverage of that candidate, perpetuating the cycle further.[2]

Until recently, it could have been that it went largely unnoticed due to the previous nature of political mistakes - a candidate in this loop would do something that breaks them out of it, and the news moves on. In Trump's case however, his mistakes are often inflammatory enough that they work to his advantage. Thus, his ratings grow, and news companies are forced to follow the profit. [3]

This phenomenon is not unique to Trump, as seen in the second graph above. Graphing all the GOP candidates' polling numbers versus their share of coverage results in a correlation of 0.96.

1.2 The Solution

This paper examines an algorithm for sorting news items in a given news cycle. It functions on one primary assumption: Over a long period of time, all the agents involved in the scenario, on average, have nearly equal standard deviation on the chronological appearance of newsworthy activity.

The algorithm works on a logarithmic principle, with hyper-exposed candidates requiring increasingly ”important” pieces of news to be represented. Thus, people are weighted down according to their respective present coverage levels.

The algorithm and the ”importance” factors shall be analyzed further in this paper, as shall the implementation in Python.

2 The News Algorithm

There are three different segments to constructing and implementing the algorithm:

- Determining which news sources are sufficiently reliable and neutral, and categorizing them.
- Aggregate articles and calculate importance factor.
- Using importance factor and candidate threshold, order news items.

2.1 Sources

Performing meta-analysis on research into quality and neutrality of news handles, they are categorized into three: Tertiary, Secondary, and Primary sources, in increasing order of reliability. They are grouped discretely as opposed to ordering on a gradient to correct for human error, due to the low amount of available research.

For the purposes of this paper, the analyses offered by Ad Fontes Media[4] and All Sides[5] were considered before the issue of a reliable news API came about, at which point the focus switched to a prototype involving static news sources: in the scenario under consideration, labelled from 1 to 9.

2.2 Importance Factor

By default, all news items have an importance factor of 10 (I). The true importance is an inverse function, i.e, the lower the factor is numerically, the more important it is. In order to calculate the true importance of an item of news, several steps are followed:

News articles are scraped from all the sources under observation, and are broken down into keywords, which are cross-referenced to identify which sources mention a specific news item. Moreover, articles are categorized into two tiers in each source according to prominence, where a ”top” headline would belong to the first tier, and others to the second tier. Percentage appearances of a news item in each category of sources are calculated.

Next, a specific category is chosen, depending on two factors: its reliability, and its percentage appearance. We choose x, y, and z as the percentage appearances of the news in Primary, Secondary, and Tertiary, and we choose the category depending on the result of largest(4*x, 2*y, z). The variable A holds x, y, or z, depending on the category chosen.

Values are assigned to the category chosen (B), and the prominence tiers in each source, as:

Primary Category - 5; Secondary Category - 3; Tertiary Category - 1

First tier - 2; Second Tier - 1

A prominence rating C is calculated as the average prominence tier of all the sources which carry the news item in that category.

Now, we use the following formula to calculate true importance:

$$importance = \frac{I}{A * B * C}$$

I = Default importance of 10,
A = Percentage appearance in the chosen category,
B = Category rating,
C = Average prominence rating

2.3 Function

By default, all people involved in the scenario have a ”threshold” factor of 10. A news item is passed through the algorithm is its importance is numerically under the threshold value. If it’s passed, the threshold is then divided by the importance factor of the news.

Thus, the algorithm weighs down candidates according to their prior exposure levels. Here is where the primary assumption holds sway: given that on average, newsworthy events are spread out chronologically, overexposure is immediately accounted for, and corrected.

Moreover, in the edge cases of breaking news that appears after a period of overexposure, the importance factor would be 1, or sufficiently close to 1, such that it is allowed to pass through, as the threshold can never drop below 1. This algorithm relies on the independent and decentralized sourcing of news from multiple reliable sources - the primary nodes of failure are in the possibility that all news outlets co-operate to present an undue image, which is indicative of larger concerns.

2.4 Updating

Every threshold level is increased by a factor of 1.0000046929 every second, or 1.5 every day. Thus, a threshold level of 2.3 would increase to 3.45 in 24 hours.

2.5 Implementation

An implementation of the given algorithm was written in Python, and contains code for the scraping of news articles, the ordering of news items from the keyword deconstruction thereof, the calculation of importance factors, the filtering and ordering of news according to threshold levels, and the function to update thresholds every second. The scraping and keyword analysis functions are unused due to the lack of reliable news APIs that cover all the analyzed sources, and therefore, static news articles were used in their place.

This implementation can be found here.

3 Conclusion

Patching flaws in high-level systems is a difficult, yet temporary fix. The issue usually stems from deeper levels, the symptoms of which are what we attempt to cover each time they pop up. Nevertheless, understanding these issues moves parallel with patching it for now.

The algorithm described in this paper is only one solution of many for the problem, and it is important to keep in mind that it is intended as a short-term measure to address immediate exploitation of the flaws, and may well be open to gaming in other aspects in the future. Solutions that work for modern systems but would predictably fail at controlling high-level actors with adverse intent are unsatisfactory.

Therefore, this algorithm may treat the symptoms for the present, and provide insight into the deeper problem whose investigation is likely to be helpful in going a long way in truly fixing the deeper problems of high-level systems.

References

[1] John Sides. “Why is Donald Trump declining in the polls? The media strike again”. *Washington Post*.

[2] John Sides. “Why is Trump surging? Blame the media.” *Washington Post*.

[3] Nerdwriter/Evan Puschak. “Why Donald Trump Is A Gift To Democracy”.

[4] Vanessa Otero. “Media Bias Chart”. *Ad Fontes Media*.

[5] All Sides. “Media Bias Ratings”.