Citation: Helmond, A. "The Algorithmization of the Hyperlink." *Computational Culture* 3 (2013). http://computationalculture.net/article/the-algorithmization-of-the-hyperlink

The Algorithmization of the Hyperlink

Anne Helmond, University of Amsterdam

Abstract

This study looks at the history of the hyperlink from a medium-specific perspective by analyzing the technical reconfiguration of the hyperlink by engines and platforms over time. Hyperlinks may be seen as having different roles belonging to specific periods, including the role of the hyperlink as a unit of navigation, a relationship marker, a reputation indicator and a currency of the web. The question here is how web devices have contributed to constituting these roles and how social media platforms have advanced the hyperlink from a navigational device into a data-rich analytical device. By following how hyperlinks have been handled by search engines and social media platforms, and in their turn have adapted to this treatment, this study traces the emergence of new link types and related linking practices. The focus is on the relations between hyperlinks, users, engines and platforms as mediated through software and in particular the process of the algorithmization of the hyperlink through short URLs by social media platforms. The important role these platforms play in the automation of hyperlinks through platform features and in the reconfiguration of the link as database call is illustrated in a case study on link sharing on Twitter.

Introduction

The hyperlink as a key natively digital object (Rogers 2013, 13) is considered to be the fabric of the web and in this role has the capacity to create relations, constitute networks and organize and rank content. Strands of hyperlink studies have been distinguished that deal with links as objects that form networks, objects that signify a particular type of relationship and the use and usability of links (De Maeyer 2011). What this study aims to contribute to these approaches to studying hyperlinks is an account of the mediating role of software in the production, uptake, processing and circulation of links by looking into the technical (re)configuration of the hyperlink over time in relation to web devices such as search engines and social media platforms.

First, I address how these devices have handled the hyperlink and how they have reconfigured the link to fit their devices. By distinguishing between 'traditional' manually created hyperlinks and hyperlinks configured by software and platforms I focus on increasing automation in the creation of hyperlinks through platform features. Social media platforms have introduced a number of alternative devices to organize relations between users, web objects and content through web activities of sharing, liking, tweeting or digging enabled by social buttons. These devices are understood as pre-configured platform links, which function as a call into the database initiating data connections with the associated platform. Next, special attention is paid to the role of social media platforms in the automatic reconfiguration of the hyperlink into the data-rich format of short URLs through a case study on link sharing on Twitter.

Following Tarleton Gillespie's (Gillespie 2010) and Cornelius Puschmann and Jean Burgess' (2013) work on platform politics, the article concludes by addressing the politics of data flows of short URLs by closely analyzing the role of the Twitter platform architecture and related third-party services in its link-sharing environment. Such an approach to understanding the reconfiguration of the hyperlink examines the mediating capacities of the platform and foregrounds the specificities of the platform itself. By following the trail of a shared hyperlink on this social media platform, one can begin to unfold the processes and actors involved in turning the hyperlink from a navigational object into an analytical device. Further, I use these findings to discuss the implications of the algorithmization of the hyperlink not only for users but also the web itself.

Through a historical and medium-specific account of the hyperlink that foregrounds its socio-technical relationship with devices, the paper revisits the political economy of linking (Walker 2002) within the era of social media. I will argue that social media platforms reassemble the existing configuration as the hyperlink enters into a process of algorithmization by shifting from a navigational object to a call into a database.

The link as technical artifact

In 'The World-Wide Web' Tim Berners-Lee et al. describe how much of the infrastructure of the web is made out of HTML (Berners-Lee et al. 1994, 78) in which the hyperlink is 'the basic hypertext construct' (W3C). The hyperlink is considered 'the basic structural element of the Internet' (Park 2003) and the 'essence of the Web' (Foot et al. 2003) and, accordingly, the fabric of the web. In a technical sense the hyperlink is 'a technological capability that enables one specific website (or webpage) to link with another' (Park 2003, 49) where it is often noted that 'a hyperlink is not only a link but has certain sociological meanings' (Hsu and Park 2011, 364). In this view, links between websites 'represent relationships between producers of Web materials' (Foot et al. 2003) and these relationships may indicate a 'politics of association' (Rogers and Ben-David 2008, 499). While links establish technical connections between websites they are also considered to organize various types of social and political relations between actors. In her overview of link studies Juliette De Maeyer discusses how hyperlinks are studied as indicators of social phenomena including authority, performance or political affiliation (De Maeyer 2013, 739–740). In short, since its inception the hyperlink has been attributed various roles beyond its function as a technical artifact.

This is in sharp contrast with its original design as made explicit by Tim Berners-Lee in his commentary on the web's architecture: 'the intention in the design of the web was that normal links should simply be references, with no implied meaning'[1] and that a link between two pages does not necessarily imply an endorsement (Berners-Lee 1997). Eszter Hargittai illustrates the hyperlink's impartiality through its technical design by describing how 'technically speaking, all hyperlinks are created equal. They can be easily inserted into any page with the simple code <a href="http://abc.xy">text or image</a>' (Hargittai 2008, 87). This means that 'a link, by itself, cannot distinguish fame from infamy' (Finkelstein 2008, 117) and that the different attributions and interpretations of hyperlinks are co-constituted by various actors on the web including users, webmasters, blog software, search engines and social media platforms, each employing them for their own distinct purpose. While Berners-Lee envisioned the link as a reference with no implicit or explicit meaning nor as representing an endorsement, devices such as search engines and social media platforms have taken up the hyperlink as an indicator to represent a variety of relations. They have also actively intervened in the ways the relations between hyperlinks, users, search engines and platforms are organized by technically reconfiguring the hyperlink to fit the purpose of the device.

This kind of intervention draws attention to the role of software as 'the invisible glue that ties it all together' (Manovich 2013, 8) in creating and mediating these linking relations. Engines and

[1] In 1997 Tim Berners-Lee distinguished between two different types of links on the web, normal hyperlinks and embedded hyperlinks. Normal hyperlinks are technically defined by the HTML properties 'A' and 'LINK' while embedded links refer to embedded objects with 'IMG' or 'OBJECT' (Berners-Lee 1997).

devices not only permeate social relations or support sociality but also contribute to structuring relations between human and non-human actors on the web (Beer 2009; Bucher 2012; Langlois, Elmer, et al. 2009; Niederer and Van Dijck 2010; Ruppert, Law, and Savage 2013). Hyperlinks are the main web-native objects creating this dynamic web of relations and web devices intervene in the assemblage by mediating the hyperlink through software and platform features. In the context of this paper, this reconfiguration of the hyperlink is traced in relation to the rise of devices on the web, first through the industrialization of the hyperlink by search engines, second through the automation of the hyperlink by (blog) software and finally through the algorithmization of the hyperlink by social media platforms.

The industrialization of the hyperlink

In the early days of the web, often referred to as Web 1.0, links were mainly created by webmasters and were considered references or connections between website A and website B (Berners-Lee 1997; Park 2003). However, this role changed over time since, 'With the initial extension of the Web, hyperlinks took on an increasing role as tools for navigation, transporting attention from place to place' (Halavais 2008, 51). With links as tools for navigation, new actors on the web such as human-edited directories started to collect these links by aggregating them into a single place as a reference for useful websites.

As the size of the web increased, other actors like search engines began to automatically index and publish these links. The automatic output of a list of links was based on matching metatags (Rogers 2006, 41), but as these types of search engines were prone to spam, a new search engine, Google, took a different approach. Google changed the idea of ranking links by also taking the link structure of the web into account. It treated the link as an authority measure, based on the academic citation index, by calculating a ranking for each link based on the weight of sites linking to it (Brin and Page 1998). The PageRank algorithm calculates the relative value of a site and in doing so, Google determined that not all links have equal value. In order to gain a higher ranking in Google, webmasters aimed to attract links from authoritative sites. This gave rise to the business of search engine optimization with its strategic link building practices and the buying and selling of links on the black link market (Walker 2002, 73). In this way, Google created an economy of links and within what has been termed 'link economy' turned the link into the currency of the web (Walker 2002; Rogers 2002). Search engines such as Google now regulate the value of links within this economy and have contributed to the '"industrialization" of the link' (Turow 2008, 3) through their automatic indexing, processing and value determination of links.[2]

This link economy of Web 1.0 was based on a relatively open web environment where webmasters and bloggers manually create links between websites and webpages, thereby creating the fabric of the web. The links are traditional one-way links and point from website A to website B where the link is displayed on the former. The link is openly visible and indexable for various search engine parties. While such practices have mainly been based on webmasters manually creating links between websites, both blogs and later social media platforms have advanced more automated forms of linking. The next section briefly addresses this increasing automation of links by software and, as a consequence, what new types of links have been introduced.

[2] The industrialization is described by Turow in relation to 'the growth of an entirely new business that measures an advertisement's success by an audience member's click on a commercial link' where the hyperlink is 'the product of a complex computer-driven formula' (Turow 2008, 3).

The automation of the hyperlink

In the early days of the web, there were no specific rules for designing or naming your link and webmasters would usually manually construct and name their hyperlinks, for example: http://mywebpage.com/aboutme.html or http://www.yahoo.com/news/sports/ and manually link to other websites.

With the increasing popularity of the web and growing number of websites in the mid 90s, systems were developed to automatically create hyperlinks between key topics (Sotomayor 1998). Scripts and the introduction of content management software also introduced automatically generated links such as http://www.nsf.gov/cgi-bin/getpub?nsf9814 and this often led to a decreased readability of links and also contributed to the lost art of reading hyperlinks. It could also involve link breakage when webmasters changed to a different script or content management software. This led to link design recommendations by Berners-Lee at W3C who made the explicit statement that 'Cool URIs don’t change' (Berners-Lee 1998).

In the late 90s a new type of website, the blog, with its characteristic reverse-chronological order, put forward a new problem in linking as early blogs displaying the latest blog post on top had no way to refer to a specific blog entry. The blog community, in collaboration with blog software developers addressed this problem with the creation of a new type of link, the permalink. By giving 'each blog entry a permanent location at which it could be referenced--a distinct URL' (Blood 2004, 54), the permalink enabled linking to the web-native unit of the blog, the blog post:

> When the Web began, the page was the *de facto* unit of measurement, and content was formatted accordingly. Online we don't need to produce content of a certain length to meet physical page-size requirements. And as the Web has matured, we've developed our own native format for writing online, a format that moves beyond the page paradigm: The *weblog*, with its smaller, more concise, unit of measurement; and the *post*, which utilizes the medium to its best advantage by proffering frequent updates and richly hyperlinked text (Hourihan 2002).

The permalink is a specifically formatted hyperlink for the medium of blogs to facilitate stable references between blog posts and it became the default output of links for blog posts in all major blog software.

Based on the permalink, blog software developers introduced a new, semi-automatic way of linking blogs with the introduction of the trackback and the pingback, which automatically notify blog B of a link received from blog A.[3] These automatic link notification systems, enabled by the default settings in blog software, created reciprocal links between blogs, making the link openly visible on both blogs. Trackbacks and pingbacks, in other words, are automated linking mechanisms making previously invisible links on the receiving end visible by displaying them underneath the blog post, usually within the comment space.

Blogs also further extended the notion of user-generated linking, previously mainly reserved to webmasters, by enabling users to leave links in the form of comments in the blog's comment space. Instead of webmasters placing links, users – and consequently spammers – could now also create links. With the opening up of the act of linking, the proliferation of links and comment spam links became a serious problem and affected the link economy.

As a response, Google announced a new link attribute (rel="nofollow") on hyperlinks in 2005 so that links with this attribute would no longer pass on value (Cutts and Shellen 2005). The attribute was widely supported by all major search engines and blog software providers and by implementing it into their software 'nofollow' became a standard attribute for all links in blog

[3] Pingbacks were developed to address the spam problems of trackbacks by automatically verifying the request and uses a different protocol than trackbacks.

comments. It was a direct intervention of a leading search engine to devalue software-created links such as pingbacks and user-generated links in the comment space.[4]

These examples of changing link types in the blogosphere show the changing configuration of actors in the political economy of links in Web 2.0 where the relations created by the proliferation of user-generated links and software-generated links and the automatic indexing of these links pose a challenge to the engines. In response, Google actively intervened in the production of hyperlinks by introducing a new link attribute to fit the device in order to prevent a (spammy) disruption of the link economy. The next section moves on to explore how social media platforms and their features have contributed to this increasing automation of the production and circulation of hyperlinks.

The pre-configuration of the hyperlink: the link as database call

Since the early 2000s the act of creating and sharing links is no longer a manual task belonging to webmasters but is increasingly being performed by software, engines, but also platforms as this section seeks to address. As Ganaele Langlois et al. have previously argued, the rise of Web 2.0 entails a shift in the actors involved in the creation and distribution of hyperlinks:

> Furthermore, whereas the production of hyperlinks in the Web 1.0, HTML-dominated environment was created by human users, hyperlinks in Web 2.0 are increasingly produced by software as tailored recommendations for videos or items of interest, suggested friends, etc. The technocultural articulations that regulate the production and circulation of hyperlinks are thus different in the Web 2.0 environment from the Web 1.0 environment, particularly with regards to the re-articulation of hyperlink protocols within other software and protocological processes (Langlois, McKelvey, et al. 2009).

[4] This may be seen in the way Google describes which types of links should receive this new attribute: 'We encourage you to use the rel="nofollow" attribute anywhere that users can add links by themselves, including within comments, trackbacks, and referrer lists. Comment areas receive the most attention, but securing every location where someone can add a link is the way to keep spammers at bay' (Cutts and Shellen 2005).

The tendency towards the previously discussed automation of links has been further fostered with the advent of sharing as 'a distributive and a communicative logic' behind participation on Web 2.0 sites and social media platforms (John 2012, 169). Sharing is seen as the 'fundamental and constitutive activity of Web 2.0' (John 2012, 167) where Web 2.0 'relies on shared objects – and avenues for circulating said objects' (Elmer and Langlois 2013, 49). The sharing logic of Web 2.0 is for instance mediated by and coded into social buttons (Gerlitz and Helmond 2013; Van Dijck 2013) which have become typical features of social media platforms to enable the easy circulation of links across different platforms. Robert Gehl argues that Web 2.0 sites thrive on aggregated content and would appear as empty frames without it: 'Without content, these sites are lifeless shells. Without it, Web 2.0 cannot work' (Gehl 2010, 42).

Many Web 2.0 sites and social media platforms prosper by making use of links and are 'as so many Web 2.0 applications seem to be, taking advantage of the openness of the Web and the underlying associations embedded in its link structure' (Nisenholtz 2008, 131). In this context it is of particular interest to look at a new type of link aggregator[5] that became very popular in Web 2.0: the social news website. These sites are of particular interest here because they have introduced new ways of automated linking enabled by social buttons (Gerlitz and Helmond 2013). Digg and Reddit are seen as the prototypical social news sites where users comment and/or vote on submitted stories or links. Users can either submit links directly on the site through a button or on an external website that automatically submits the link to the platform. I detail the development of this automatic submission process within the context of link sharing by focusing on Digg to describe how pre-configured platform links have become embedded in social buttons to enable automated sharing practices between websites and platforms by turning the link into a device to automatically submit and retrieve data.

[5] These sites may be seen as precursors of earlier link aggregation sites, or group blogs, such as the collaborative filtering site Metafilter and tech news and discussion site Slashdot.

Initially users could only submit and vote on stories from within the Digg website itself where the site would specifically treat the submission of a story, a link, as a vote. Webmasters and bloggers encouraged users to submit their own stories by directly linking to the Digg stories submit page, but there was no way to submit or Digg a story from an external website. In July 2006, Digg announced a new 'code push' to directly integrate Digg into a website (Rose 2006). This new integration functioned in two ways: Firstly, it allowed users to directly 'Submit To Digg' from the website they wanted to submit, through the 'Digg Story Button' — instead of being redirected to the submissions page – and, secondly, it allowed webmasters to display the number of Diggs for a story, in other words the number of votes for a link. The first step enabled automated link submissions to Digg by 'pre-populating the submission form with a title, description, and topic' in the following manner: 'http://digg.com/submit?phase=2&url=www.UniqueURL.com&title=StoryTitle&bodytext=StoryDescription&topic=YourSelectedTopic' (Rose 2006). This pre-populated link could be used as the anchor of a Digg image thereby creating a social bookmarking icon that functioned as an external submission mechanism. In a second step, JavaScript could be used to display the number of Diggs for a particular URL and to allow users to Digg or vote on the story directly from the website:

```
<script>
digg_url = 'URLOFSTORY';
</script>
<script src="http://digg.com/api/diggthis.js"></script> (Rose 2006)
```

Combined with a Digg icon this created a social button[6] that, on the one hand, automates link submissions from external websites and, on the other hand, decentralizes Digg features by

[6] Such icons or buttons are often also referred to as social bookmarking icons. This is derived from their initial function to save or share a page on a platform such as Delicious or Digg. Through a deeper integration with the connected platform the functionality of these buttons has changed over the years by also allowing showing the

providing Digg-functionality on external websites. This functionality includes displaying the number of Diggs, or votes, for a particular link and the ability to directly vote on the story if it has already been previously submitted to the site. This process, as enabled by social buttons, can be seen as an important step in the automatic sending and retrieval – or exchange – of data and functionality between websites and platforms (Gerlitz and Helmond 2013), now commonly achieved through APIs, that is Application Programming Interfaces. This is also explicitly stated within the Digg documentation, which announced the new submission mechanism as a way for third-parties to interact with Digg's data in absence of an API:

> This document outlines the current way we allow partners and third parties to put "Digg This" and "Submit to Digg" links on their site. Our strategy is to bring both types of links to the story submission URL at Digg, which will allow a user to submit the story if it doesn't exist. If it happens to already exist, they will be able to see the story and digg it on our site. The formal API alternative is still in development, and when complete, will allow more direct integration with Digg's data (Digg 2006).

With this new 'External Story Submission Process' Digg put forward a new type of linking practice: *pre-configured linking*, where the link is automatically configured for platform submission when clicking the button. The issue here concerns how this practice has not only contributed to the automation of the hyperlink, but also to the algorithmization of the hyperlink by setting up channels for data exchange. This is seen in relation to the increasing modularity of the web, often associated with Web 2.0 as an architecture of modular elements heavily relying on APIs for the exchange of data (Langlois, McKelvey, et al. 2009) and the rise of social media platforms within this infrastructure. The term 'platform' could be seen as having a computational, architectural, figurative and political meaning, implying different things for different actors, as put forward by Tarleton Gillespie in his discourse analysis on the politics of platforms (Gillespie 2010). In this paper I follow the computational meaning of a platform as defined by Marc Andreessen to analyze how platforms create an 'architecture of assembly' with modular components to disperse and exchange platform data and functionality (Ullrich et al. 2008):

number of saves or votes or shares or other interactivity related to the page or link. The icons or buttons have since often also been referred to as share buttons or social buttons.

> Definitionally, a "platform" is a system that can be reprogrammed and therefore customized by outside developers – users – and in that way, adapted to countless needs and niches that the platform's original developers could not have possibly contemplated, much less had time to accommodate (Andreessen in: Bogost and Montfort 2009).

In this sense, a social media website becomes a platform once it can be reprogrammed, these days typically achieved by providing an API allowing the structured exchange of data and functionality between websites and services. In Web 2.0, or 'the web as platform' (O'Reilly 2007), websites have increasingly become database applications where links in social buttons function as queries into the platform's database and as connection initiators. Technically, the devices that enable the automatic sharing of links are not hyperlinks, rather they are API calls into the platform's database that enable data exchanges, where the hyperlink itself is just one of the many fields within the database.

Due to this particular setup, social buttons allow for a different type of linking than the previously described mechanisms. Social buttons do not create a link between two websites or blogs, but rather, between websites and platforms, thereby automatically recentralizing all links to the platform (Gerlitz and Helmond 2013). The platform then aggregates these links and uses them as a voting mechanism in which each link has equal value. This stands in sharp contrast with search engines, where links have different values, even within the same site, and can also be deprived of their value, as in the case of the comment space. However, links shared on platforms do not have equal value in terms of visibility, as some links may only be visible from within the platform itself and to different platform populations.

It may be argued that the social buttons and the associated ranking of links on their platforms have taken up the empty space of the value of the user-generated link and the software-generated link after Google rendered these links worthless with the 'nofollow' attribute

(Weltevrede 2011). Social media platforms as actors within the link economy create their own value from links outside the 'nofollow' attribute since while links on platforms have a 'nofollow' attribute, which means they do not pass on value for rankings in search engines, they do contribute to the value on the platforms themselves. In this sense, each platform has its own value feature: the Like, the (re)Tweet, the +1.

Social buttons mediate link sharing in particular ways by automating the practice of linking through the pre-configuration of links, while at the same time introducing themselves as intermediaries in link creation and circulation. Platforms emerge as an interface between users, webmasters and search engines and 'arise as sites of articulations between a diverse range of processes and actors' (Langlois, McKelvey, et al. 2009). For the political economy of linking in the era of social media, platforms become important actors in the production and distribution of links while at the same time regulating access to these links for engines.

Due to the proliferation of links by the automated linking practices of social buttons and due to the closed character of particular social media platforms, the engines cannot index all links. But, according to Matt Cutts, head of Google's Webspam team, if Google can crawl Twitter and Facebook links it will use them as a social signal in their web search ranking.[7] While some engines enter direct partnerships with social media platforms,[8] the platforms are establishing themselves as independent new players in the economy of links by building on the industrialization and automation of the link. Besides putting forward more automated forms of linking, social media platforms have repurposed an already existing new type of link, the short URL, as an automated device for information harvesting.

[7] On top of that, Google also takes the reputation of the author on Twitter and Facebook into account. See http://www.youtube.com/watch?v=ofhwPC-5Ub4.

[8] Google & Twitter had a partnership until 2011 and currently Yahoo! & Twitter and Bing & Facebook have engaged in partnerships.

Shortened links as data-rich URLs

The idea of URL shortening goes back to 2001 when the service makeashorterlink.com was launched to transform long URLs into shorter ones as an answer to long links breaking – and becoming unclickable – in email clients.[9] The mainstream success of URL shortening services is often attributed to Twitter, because of the platform's 140-character limit (Antoniades et al. 2011, 715). In a medium where every character counts, users started to employ URL shortening services to save characters by shortening their links. These URL shortening services (USS) do not technically change the link but create an alias for the link, the short URL, which then redirects to the original long URL. When the short URL is requested the USS issues a 'HTTP/1.1 301 Moved Permanently'[10] status code response that tells the browser that the location of the short URL has been moved permanently and redirects the user to the location of the original long URL. Short URL redirection happens largely unnoticed by users but the technique, which is currently used by more than 600 distinct USS (Maggi et al. 2013, 861), adds an extra layer of indirection to the web by creating an additional hyperlink that needs to be resolved (Schachter 2009).

This new layer of short URLs can be analyzed in relation to the algorithmization of the hyperlink by social media platforms as prime contributors to this production of an additional element of infrastructure. It is important to distinguish between two types of shorteners to analyze this new layer of short URLs to discuss its implications for the various actors on the web. Federico Maggi et al. (2013) differentiate between general URL shortening services such as tinyurl.com, bit.ly and ow.ly and site-specific URL shortening services such as flic.kr (Flickr), youtu.be (YouTube), fb.me (Facebook), t.co (Twitter) and goo.gl (Google). The general shorteners, popularized by

[9] See http://web.archive.org/web/20010713184016/http://makeashorterlink.com/about.php. Also, a patent filed in 2000 mentions the idea of using 'shorthand URLs' (Megiddo and McCurley 2005).

[10] While there are other types of redirection 'HTTP/1.1 301 Moved Permanently' is the most commonly used redirect in USS.

TinyURL, allow you to convert any long URL into a short URL after which the short URL can be easily shared across various platforms. The second type of shorteners, the site-specific USS, are connected to the website itself and often tied into website features.[11] Social media platforms in particular use site-specific USS to integrate them into their platform-specific sharing features. In these cases, sharing content from a website or social media platform using social buttons, which through their embedded pre-configured linking practices enable the easy sharing of content outside the boundaries of the platform, may automatically produce short URLs. This is the case, for example, when you share a video from YouTube through the 'Share this video' option on the platform. In this scenario, YouTube automatically produces a short YouTube URL to share: http://youtu.be/Yv73cRpbQaE. Or, if you share an article from The Huffington Post on Twitter using the 'tweet' button a Twitter pop-up appears which contains an automatically produced tweet with the article's title and Huffington Post short URL: http://huff.to/xs4lYU. This mechanism is similar to the previously discussed automated link submission from Digg, except that the pre-configured link produced is a short URL.

In this sense, sharing as the essential activity of Web 2.0 (John 2012) is mediated by platform-specific features such as social buttons, which may automatically create platform-specific short URLs.[12] Users, websites, platforms and USS together contribute to the additional layer in the web's infrastructure by producing short URLs.

[11] However, not all site-specific USS are run by the websites themselves. Rather, some use a general USS such as bit.ly that offers a custom URL service. Bit.ly is one of the main providers of custom short URLs for websites and they power over 10.000 custom short domains including Foursquare (4sq.com), The New York Times (nyti.ms) and The Huffington Post (huff.to), see http://blog.bitly.com/post/284009728/announcing-bit-ly-pro. Since everyone can create a custom short domain powered by bit.ly it has become increasingly difficult to distinguish between general USS and site-specific USS.

[12] Not all sharing features distribute a site-specific short URL and it also depends on whether a link is shared from the web or a mobile platform.

The proliferation of links through data-rich short URLs

Sharing links may automatically produce short URLs and in doing so create an extra alias for the existing link. On top of that, most USS create a *unique* short URL for each long URL submitted or shared. This means that if 20 people each share the same article from the Huffington Post to Twitter using the 'tweet' button 20 new unique huff.to URLs and 20 new unique t.co URLs are automatically produced. The shared Huffington Post article no longer has a single URL that points to its location on the web, but now exists under an additional 20 unique huff.to URLs and 20 unique t.co URLs which all point to, or rather redirect to, the original URL. The sharing features and infrastructure of the Twitter platform contribute to a proliferation of links in order to track the further sharing of these links within and outside of its platform by gathering statistics for the short URL.

The ability to track analytics for short URLs was conceived by a second generation of general USS who added new value to their service by not only shortening the URL but also by providing statistics for the shortened link. They transformed the short URL into a data-rich hyperlink suitable for analytical purposes. The hyperlink as a short URL is no longer a navigational tool but carries information on the clicks and other (user) data related to it. The biggest general USS, Bit.ly, offers metrics including the number of clicks on the bitl.y link, the date and time the link was clicked, the country the link was clicked from and where the link was shared from, e.g. Twitter, Facebook, Google+, email etc.[13] Besides providing data for each unique short URL these general USS also offer aggregated data for all related short URLs that have been created for the same long URL: many short URLs may be resolved to one single global shortened URL to keep track of a cumulative count of statistics for the original long URL (Chhabra et al. 2011, 95).

[13] Link statistics can be accessed on the Bit.ly website by adding a '+' at the end of any bit.ly link or through the Bit.ly API: http://dev.bitly.com/link_metrics.html.

This capacity of the short URL as an analytical device may provide websites with valuable information about the popularity and spread of their content. Indeed, instead of using a general USS for statistics, many websites (and in particular social media platforms) have started to implement their own site-specific USS. This has four main advantages: first, websites can use a custom shortener to create vanity URLs (e.g. Ars Technica = arst.ch); second, they are not dependent on an external service to provide them with statistics; third, by not relying on an external service to resolve short URL aliases – with the risk that if the USS goes down the shortened links created by the USS will no longer resolve and thus break – the short URL will work as long as the website it refers to is still online (Winer 2009); and finally, by routing all links through an internal USS they can be connected to a user profile on the platform.

At the end of 2009, two major platforms, Facebook and Google, quietly introduced their own site-specific USS by embedding them into their services (Sterling 2009) to track the spread of their links across the web.

The algorithmization of the hyperlink

So far I have discussed the transformation of the hyperlink from a navigational device into an automated analytical device through the proliferation of a new type of hyperlink, the data-rich short URL. I now move on to the role of platform infrastructure to analyze the changing role of the hyperlink within social media, by focusing on Twitter. Attention is drawn to the mediation of links, which entails a reconfiguration of the hyperlink into an algorithmic device to fit the platform.

Initially Twitter did not have its own URL shortener and many users relied on general USS to shorten their links before sharing them onto the platform. In 2008, Twitter integrated TinyURL as its default shortener to automatically shorten all links shared on the platform, but later replaced it with Bit.ly, which offered metrics and better uptime by comparison (Kirkpatrick 2009). In June 2010, Twitter announced it had been testing its own URL shortener by wrapping shared links in private messages with a twt.tl URL (Garrett 2010). A year later, the company officially rolled out its own internal link shortening service across the whole platform with t.co (Twitter 2011). T.co is Twitter's so-called 'link wrapper', a service that 'wraps' all the links shared on the platform with its own platform-specific t.co short URL. In other words, Twitter routes all links through its own t.co link service and in the process shortens these links to a 22-character t.co URL and wraps them into a data-rich layer. This means that even already shortened URLs, or URLs shorter than 22-characters, are transformed into a 22-character t.co URL. This process is referred to as link wrapping because while technically Twitter changes the 'href' of the original URL into a t.co URL it will usually not display this t.co URL; rather, it keeps the full URL in the link code created with HTML and displays a shortened version of the original URL in a semantically comprehensible way.[14]

Twitter developer Taylor Singletary explicitly states that t.co is not simply a URL shortener,[15] but a URL wrapper and that Twitter wraps all links for two purposes: first, to protect its users from malicious links to phishing or malware websites and second, to understand how users engage with shared links (Twitter 2011). By rerouting all links through their platform, it can detect spam links by matching them against a list of known malicious links and, subsequently, warn for or even block potentially harmful links. Besides providing a phishing protection mechanism the link wrapping creates a short URL, which can be used to gather statistics on the link:

[14] "url": http://t.co/9YEaNmtd

"display_url": "huffingtonpost.com/2012/01/14/cos..."

"expanded_url": http://www.huffingtonpost.com/2012/01/14/costa-concordia-disaster-_n_1206167.html.

[15] 'Shortening URLs" isn't a primary focus or "purpose" of t.co. Wrapping URLs for safe redirection & abuse prevention is the primary goal of t.co, regardless of a URL's original length' (Singletary 2011).

> Twitter may keep track of how you interact with links across our Services, including our email notifications, third-party services, and client applications, by redirecting clicks or through other means. We do this to help improve our Services, to provide more relevant advertising, and to be able to share aggregate click statistics such as how many times a particular link was clicked on ("Twitter Privacy Policy" 2012).

The hyperlink has become an analytical device in the form of a short URL used to gather information about link sharing behavior as each link shared on the platform can be traced and connected back to an individual user profile. Twitter's practice of automatic link wrapping creates a data-rich short URL that becomes one of the many fields in the platform's database and may be used for different purposes, including metrics and analytics:

> In addition to a better user experience and increased safety, routing links through this service will eventually contribute to the metrics behind our Promoted Tweets platform and provide an important quality signal for our Resonance algorithm – the way we determine if a Tweet is relevant and interesting to users. We are also looking to provide services that make use of this data, an example would be analytics within our eventual commercial accounts service (Garrett 2010).

By automatically wrapping links in tweets with a t.co URL, Twitter makes this shared data on its platform 'algorithm ready' (Gillespie forthcoming) by reconfiguring the hyperlink to fit the platform. The automatic processing of the hyperlink and its reconfiguration into an analytical device in order to become part of an algorithmic system is what I refer to as the algorithmization of the hyperlink. Hyperlinks in tweets may serve as signals in various algorithms that structure, filter and recommend content on the Twitter platform including the relevance algorithm behind Twitter Search, the popularity algorithm behind Trending Topics and the personalization algorithm behind the Discover tab to find new and relevant content:

> To generate the stories that are based on your social graph and that we believe are most interesting to you, we first use Cassovary, our graph processing library, to identify your connections and rank them according to how strong and important those connections are to you.
>
> Once we have that network, we use Twitter's flexible search engine to find URLs that have been shared by that circle of people. Those links are converted into stories that we'll display, alongside other stories, in the Discover tab. Before displaying them, a final ranking pass re-ranks stories according to how many people have tweeted about them and how important those people are in relation to you (Twitter 2012).

David Beer discusses how 'the activities of content generation and participation of Web 2.0 feed into relational databases and are then used to sort, filter and discriminate in automated ways and without users knowledge' (Beer 2009, 998) where these activities, including link sharing, are mediated by the platform's infrastructure in anticipation of being fed into an algorithm. Links on Twitter become part of multiple algorithms within the platform but they also become part of multiple databases and algorithms outside of the platform. These include 'Twitter Certified Products' partnerships such as SocialFlow using links in its AttentionScore™ algorithm and Topsy's link trending and relevance algorithm. But links on Twitter may have also entered multiple relations with engines, devices and other platforms before being posted onto Twitter. These include sharing practices using social buttons and cross-posting practices where users post a link on one platform which then automatically posts it to other platforms. This may be achieved by connecting platforms in the platform's settings, e.g. 'Link Your Facebook Profile to Twitter'[16] or by using external tools such as If This Then That.[17] Links passing through various channels become part of the larger link-sharing environment on the web where each device or platform may reconfigure the link in order to fit their medium by making it 'algorithm ready' (Gillespie forthcoming).

The practice of link sharing on Twitter, in relation to its larger link-sharing environment on the web, can be analyzed from the perspective of the link itself by looking at the role of the platform architecture, related third-party services and other actors involved in the production, proliferation and circulation of hyperlinks to address the politics of data flows of short URLs within social media. The following section of the paper develops this analysis by means of a specific case study.

[16] See https://www.facebook.com/twitter/.

[17] The service If This Then That 'Put the internet to work for you' lets you connect web services and social media platforms with each other using APIs. See https://ifttt.com.

The Social Life of a URL shared on Twitter

Here, I follow the path of a single URL to draw attention to the reconfiguration of the hyperlink by software and platforms in the practice of link sharing on Twitter. It takes a medium-specific approach by 'following the medium' (Rogers 2013, 25) to see how the platform handles the natively digital object of the hyperlink and it reuses those techniques to analyze the actors within the political economy of linking in social media. I propose a method for following shared links on Twitter in order to trace part of the larger link-sharing environment of the social web and the actors involved in the creation, proliferation and distribution of these links.

To outline this method, I focus on one particular link shared on Twitter, a Huffington Post article on the Costa Concordia Disaster with the following URL: http://www.huffingtonpost.com/2012/01/14/costa-concordia-disaster-_n_1206167.html.[18] Because Twitter creates a unique t.co for every link shared on its platform we first need to detect all the instances of the Huffington Post URL on Twitter. This immediately poses a challenge because the platform itself does not offer a straightforward way to retrieve all the mentions of the link within the platform. The Twitter Search web interface allows you to search for a URL, but does not return a count for the number of tweets the URL is referenced in nor provide historical data for 'old' shared links. The Twitter APIs, in this case the Search API and Streaming API, are the industry-preferred methods to get access to Twitter data, but while the Search API allows you to query for a URL it only provides an index of recent Tweets (6-9 days) which is further limited because 'not all Tweets will be indexed or made available via the search interface.'[19] The Streaming API limits requests with too many parameters making it unable to

[18] Initially the data was gathered manually from Topsy on January 16, 2012. It was redone on April 18, 2013 by gathering data from the Topsy API in order to expand the application of the proposed method to historical data.
[19] See https://dev.twitter.com/docs/using-search.

track a full URL.[20] Moreover, because it is designed 'for receiving tweets and events in real-time,'[21] it does not serve past tweets, making both Twitter APIs unsuitable entry points for the analysis.

Retrieving all mentions for a specific URL can be achieved with the external service Topsy, a certified Twitter partner offering 'instant access to realtime and multi-year analyses from the world's largest index of public Tweets.'[22] Topsy is a 'realtime social search engine' that allows you to search for a topic or URL mentioned on Twitter. Most importantly, in contrast to Twitter itself, Topsy provides historical data making it a suitable entry point for analysis. The Topsy API allows URL requests to be made and returns a list of unique tweets referencing the requested URL, called Twitter Trackbacks.[23]

The Topsy API returned 172 unique Twitter Trackbacks, tweets containing the specified URL, for the Huffington Post URL.[24] In this case study I am particularly interested in tracing the software mediation from the actors involved in link sharing, so in a next step I extracted all links from the tweets using the Harvester tool which extracts URLs from text.[25] The tool extracted 147 unique t.co URLs from the 172 tweets containing the URL meaning that there are a 147 unique links on Twitter that all refer to the same article on the Huffington Post. These 147 URLs returned by the Topsy API are all t.co links because Twitter automatically transforms all

[20] When using a full URL as the input parameter the API returns an error that the track keyword (the URL) is too long. A canonicalized domain (example.com) is accepted however. See https://dev.twitter.com/docs/streaming-apis/parameters#track.

[21] See https://dev.twitter.com/discussions/6971.

[22] See https://dev.twitter.com/programs/twitter-certified-products/topsy and http://topsy.com/.

[23] On Topsy a trackback is a reference to a URL by an author. A tweet with an URL is a Twitter Trackback. This is slightly more expansive usage of the term compared to its original meaning. The Otter API provides the list of trackbacks for any URL that is in the Topsy index.' https://code.google.com/p/otterapi/wiki/Glossary.

[24] See http://otter.topsy.com/stats.json?url=http://www.huffingtonpost.com/2012/01/14/costa-concordia-disaster-_n_1206167.html&apikey=secret.].

[25] See https://tools.digitalmethods.net/beta/harvestUrls/.

links passing through its platform into t.co links using the t.co link wrapper. The single Huffington Post URL has now been proliferated into 147 unique URLs by the platform.

In the next step, I followed the path of all 147 unique t.co URLs to trace the origin of the link. By following the URL, or more specifically its redirect chain, we can detect which channels, other than the Twitter platform infrastructure, the link has passed through before ending up on Twitter.

So, in a third step, I followed the redirect chain of all these t.co links to see where they resolved to. Since t.co links, and other types of short URLs, are redirects we can use cURL[26] to follow this redirection path. CURL can follow HTTP redirects of (shortened) URLs by iteratively requesting the server's HTTP location header until the final destination is returned. The output of the header shows the path of redirection (see Appendix 1.) where the input point is a single unique t.co.

In a fourth step, the redirection paths of all 147 t.co URLs were written down in a spreadsheet. This sheet with redirections was exported as a .csv file and transformed into a Gephi file using the Table2Net tool.[27] In a fifth step, all redirections were coded in Gephi with the name of their URL shortener and UTM link tag before removing these and other tags from the URLs.[28] The connections between the links, the paths of redirects, were then analyzed and visualized using Gephi. Illustration 1 shows the t.co network of the single Huffington Post article URL.

[26] 'Curl is a command line tool for transferring data with URL syntax' See http://curl.haxx.se/.
[27] See http://tools.medialab.sciences-po.fr/table2net/.
[28] UTM (Urchin Tracking Module) are tags that can be added to an URL to track extra information about the link and may include the source of the link (the referrer, eg Google or twitterfeed) and the medium (email, newsletter, Twitter). See https://support.google.com/analytics/answer/1033863.

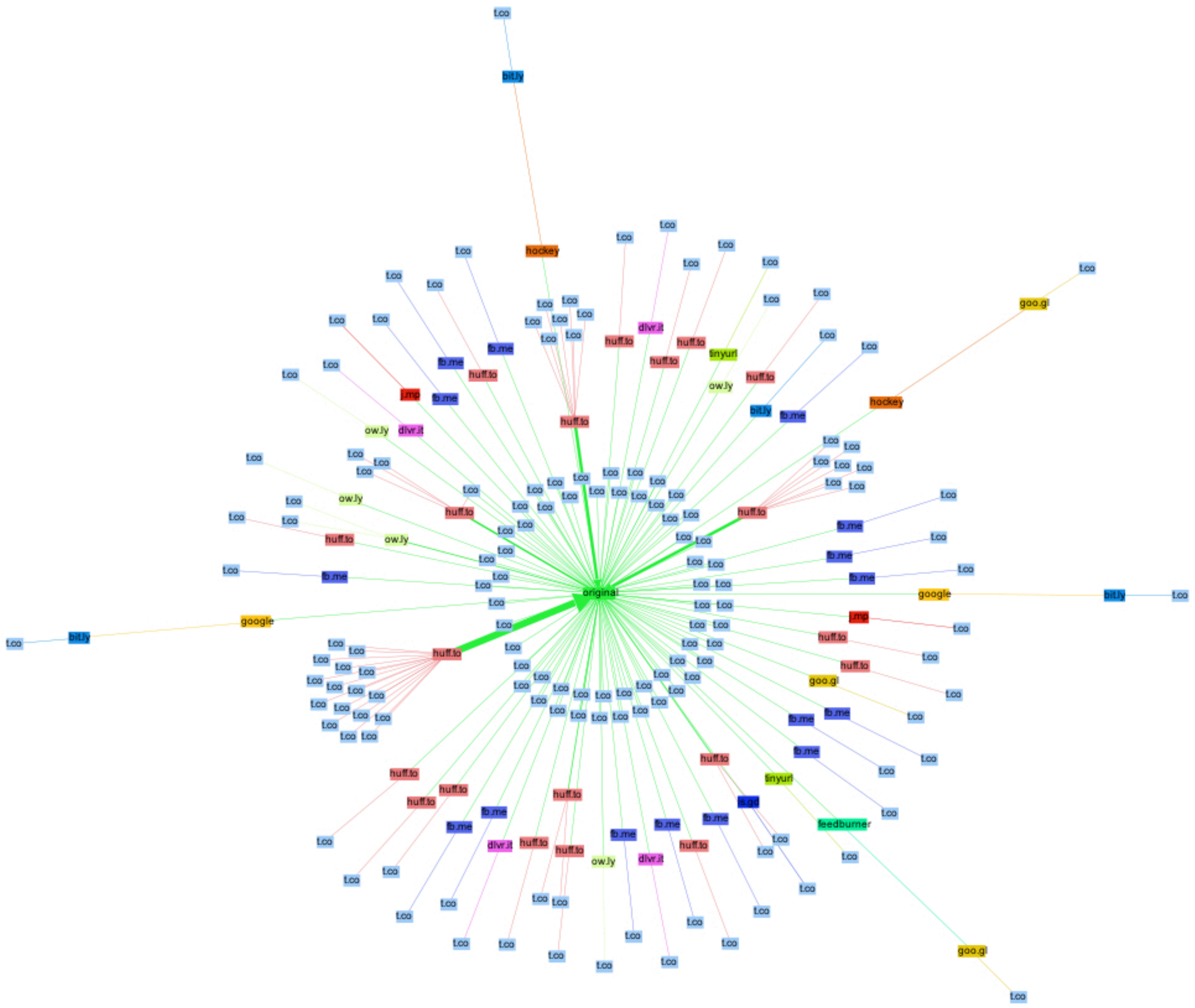

Illustration 1: The hyperlink network of a single URL

The visualization shows the network of a single link shared on Twitter and depicts the mediating capacities of devices such as engines and social media platforms.[29]

First, it shows the proliferation of the hyperlink as an effect of its reconfiguration into a data-rich short URL. There are 211 nodes (including the original Huffington Post URL) in the graph, which is more than the 147 unique t.co links that have been produced by the Twitter platform architecture. This points us to the existence of other actors involved in the proliferation of the

[29] It is a different type of visualization than normally produced as the output of hyperlink network analysis because it does not show the links between different websites but rather the network of redirection paths of single URL.

hyperlink that are creating unique URLs for the original URL when handling the link. In this case, I have discovered the following new actors: 20 huff.to (Huffington Post), 16 fb.me (Facebook), 5 ow.ly (Hootsuite), 4 bit.ly (Bitly), 4 dlvr.it (DlvrIt), 3 goo.gl, 2 google, 1 feedburner (Google), 2 tinyurl (TinyURL), 2 j.mp (Bitly), 2 hockey (spam), 1 is.gd (Is.gd).

The network diagram displayed in Illustration 1 is a visual representation of a directed graph, that is, 'every link has a source and a target' (Rieder 2013) and shows the redirect chain of each t.co. In this small sample only 57 of all 147 t.co links, depicted in the center of the visualization, directly resolve to the Huffington Post URL. This means that the other links pass through one or more channels *other* than the Twitter architecture before being posted on the platform. This may happen, for example, if the link is shared from an external platform using social buttons, or if the link is shortened using a general USS, or if the link is cross-posted from another platform. I briefly discuss each of these scenarios in relation to the new links found as they point to the various actors involved in mediating distinct sharing practices on Twitter and in creating a new layer of data-rich short URLs before moving onto the implications of this new layer.

The huff.to URL is a platform-specific short URL belonging to the Huffington Post and is automatically produced through pre-configured linking by using the 'tweet' button next to the article. Fb.me is Facebook's platform-specific shortener and is mostly used in the mobile interface indicating that these links have been shared from Facebook's mobile interface to Twitter. By shortening the link Facebook using fb.me can derive value from it by tracing it across the web and feed back these link metrics into its own platform for further use. Ow.ly, bit.ly, tinyurl, j.mp and is.gd are general URL shorteners that can be used before sharing the link on Twitter to gather link statistics but they may also be embedded in third-party Twitter applications as default shorteners. Dlvr.it is a web service to automatically cross-post your new blog content to Twitter, Facebook, Google+ and LinkedIn, and allows you to schedule your posts.[30] Goo.gl is Google's general URL shortener and is also integrated into several Google products including Feedburner for automatically posting your new blog posts to Twitter. Two of the longest redirect chains, where the link passes through three distinct actors, are spam links that have been shortened by several URL shorteners in order to obfuscate the URL by creating long redirect chains (Lee and Kim 2013).

Analyzing the redirect chains of the unique short URLs created by the Twitter platform provides an entry point into the role of software in the reconfiguration of the hyperlink throughout the link-sharing environment of Twitter. This environment can be seen as a 'sociotechnical system' defined by 'the intricate collaboration between human users and automated content agents' (Niederer and Van Dijck 2010, 1368) where link sharing is enabled by (semi-) automated devices such as Tweet and share buttons, scheduling software, cross-posting services and other third-party applications. Within this system, each actor in the redirect chain reconfigures the hyperlink to fit their platform where many transform it into a data-rich short URL to derive value from the associated data flows. In other words, these short URLs are created to enable a data connection between the user, the shared link and platform database to track the link and related statistics. The practice of link sharing on social media platforms automatically routes the link through the platform that renders the link, and to a certain extent the user, into a traceable object. As such, this case study unveils a particular strand of actors that could be added to the Twitter data ecosystem described by Puschmann and Burgess (2013): the intermediaries of link sharing, and in particular, URL shorteners and their politics of data flows. These intermediaries derive value from data flows related to the data-rich short URL such as detailed link statistics, enabled by

[30] To complicate the sharing landscape even further dlvr.it also allows you to use bit.ly as a default shortener instead of dlvr.it making it difficult to establish the actual source.

cookies.[31] In the case of bit.ly, for example, this 'information includes, but is not limited to: (i) the IP address and physical location of the devices accessing the shortened URL; (ii) the referring websites or services; (iii) the time and date of each access; and (iv) information about sharing of the shortened URL on Third Party Services such as Twitter and Facebook.'[32]

Following the web-native object of the hyperlink on a platform has revealed numerous actors that set themselves as intermediaries to track links and associated user data within the link sharing environments of social media. This case study illustrates how the link significantly changed from a navigational device into an analytical device through the automatic reconfiguration by social media platforms. For these platforms, a good hyperlink not only points from A to B, but it is also data-rich by establishing a connection with the underlying platform database.

The implications of a web of short URLs

Demetris Antoniades et. al. describe how 'URL shortening has evolved into one of the main practices for the easy dissemination and sharing of URLs,' (2011, 715) but they also point out that the short URL has hitherto remained an understudied object. They want to put the study of short URLs on the agenda since their usage in social networks and social media platforms is rapidly growing; indeed, they account for an increasingly significant amount of web traffic and are becoming a critical part of the web's infrastructure (Antoniades et al. 2011). According to their work, short URLs reflect an 'alternative web' which is created and consumed by a particular community of users on social network sites and social media platforms and this web reflects their interests (2011, 112). In this article, I focused on how users, software and platforms

[31] See https://bitly.com/pages/privacy.

[32] See https://bitly.com/pages/privacy.

collaboratively and automatically create this alternative web of short URLs through the reconfiguration of the hyperlink in the practice of link sharing.

To conclude, I would like to address the implications of this new infrastructural element, which may be seen as part of 'the core infrastructure of the social web' (Mattlemay 2011). First, the practice of automatic link shortening adds an extra layer to the web's infrastructure based on centralized hubs (Schachter 2009; Winer 2009). This means that if the USS goes down or the service gets disrupted, then all shortened links from that domain can no longer be resolved to the longer link, which causes link breakage. The Internet Archive and The Archive Team (URLte.am) have both addressed the vulnerability of this new infrastructure with short URL archival initiatives. The Internet Archive has a project titled 301works.org, 'an independent service for archiving URL mappings,' which collects lists of short URLs aliases and their corresponding long URL so that the Internet Archive can continue to resolve the short URLs in case the participating USS closes down. This initiative requires the active participation and permission of USS which is why the Archive Team, who claim that short URLs 'pose a serious threat to the internet's integrity,' have taken a more aggressive approach by scraping USS to create backups and release those backups as torrent files.[33] Another vulnerability comes from URL shorteners being registered on foreign top-level domains (t.co = Colombia, bit.ly = Libya) which comes with potential legal and censorship issues (Johnson, Arthur, and Halliday 2010).

The extra layer of short URLs also slows down the web because redirects impose additional data requests that need to be resolved ("URL Shorteners Make the Web Substantially Slower; Facebooks' Fb.me Is Slowest" 2010). In October 2010 alone, the biggest general USS, Bitly, handled over 8 billion redirects. A year later, it entered a relationship with Verisign, which operates critical parts of the Internet by running two of the 13 root DNS servers (Mattlemay

[33] See http://urlte.am/.

2011), to ensure fast and reliable short URLS (Giles 2011). This partnership also included a data-sharing agreement between Verisign and Bitly, whereby 'Verisign's data could add an awareness of activity outside the social sites where Bitly links are used' (Giles 2011). Within this arrangement, moreover, it is pointed out that 'an average user interacts more than 30 times a day with Verisign's infrastructure.'[34]

A second set of related concerns involves tracking user behavior through short URLs and the creation of user profiles by USS by setting cookies (Neumann, Barnickel, and Meyer 2010). This data may be combined with other sources through data-sharing agreements and data may licensed or sold to other parties, depending on the Terms of Service of the USS. Another privacy concern is that not all users are aware that shortened links are publicly accessible and that secret URLs may be found through enumeration or may be indexed by search engines (Neumann, Barnickel, and Meyer 2010). Besides these privacy implications, researchers also point to numerous security implications including the hacking of USS to redirect links to malicious sites or the use of USS in phishing attacks (Chhabra et al. 2011).

Finally, and to return to the changing role of the hyperlink from a navigational device to a database call, one can see that the practice of link sharing as a 'fundamental and constitutive activity of Web 2.0' (John 2012, 1) has become mediated by platforms. These platforms are reconfiguring the hyperlink to fit their medium by transforming the link into a data-rich URL in order to make the data platform and 'algorithm ready' (Gillespie forthcoming). In this process, they are automatically creating an additional layer of short URLs through the proliferation of hyperlinks by the platform infrastructure. This layer creates additional central hubs on the web, URL shortening services, which not only slow down the web or cause link breakage when the service goes down but may also track users through the practice of link sharing. The mediation

[34] See http://www.verisigninc.com/en_US/why-verisign/education-resources/how-dns-works/index.xhtml

of this practice by platforms in the form of link wrapping and URL shortening allows various actors to participate in this new infrastructure and may gain different forms of data, analytics and value for their own platforms.

Acknowledgements

I would like to thank my colleagues Carolin Gerlitz, Richard Rogers, Bernhard Rieder, Michael Dieter and David Nieborg and the two anonymous reviewers for their valuable feedback and comments. Thank you Bernhard for writing me a script that follows URLs and outputs the HTTP header and redirect information.

About the author

Anne Helmond is a PhD candidate with the Digital Methods Initiative, the PhD program in New Media & Digital Culture at the Department of Media Studies, University of Amsterdam. In her research she focuses on software-engine relations in the blogosphere, cross-syndication politics in social media and data flows between web platforms. Her research interests include digital methods, software studies, platform studies, social media and syndication protocols. She also teaches new media courses in the Media Studies department.

Bibliography

Antoniades, Demetris, Iasonas Polakis, Georgios Kontaxis, Elias Athanasopoulos, Sotiris Ioannidis, Evangelos P. Markatos, and Thomas Karagiannis. 2011. "We. B: The Web of Short URLs." In *Proceedings of the 20th International Conference on World Wide Web*, 715–724. http://dl.acm.org/citation.cfm?id=1963505.

Beer, David. 2009. "Power through the Algorithm? Participatory Web Cultures and the Technological Unconscious." *New Media & Society* 11 (6) (September 1): 985–1002. doi:10.1177/1461444809336551. http://nms.sagepub.com/content/11/6/985.

Berners-Lee, Tim. 1997. "Links and Law." *Commentary on Web Architecture*. http://www.w3.org/DesignIssues/LinkLaw.html.

———. 1998. "Cool URIs Don't Change." *W3C*. http://www.w3.org/Provider/Style/URI.html.

Berners-Lee, Tim, Robert Cailliau, Ari Luotonen, Henrik Frystyk Nielsen, and Arthur Secret. 1994. "The World-Wide Web." *Commun. ACM* 37 (8) (August): 76–82. doi:10.1145/179606.179671. http://doi.acm.org/10.1145/179606.179671.

Blood, Rebecca. 2004. "How Blogging Software Reshapes the Online Community." *Communications of the ACM* 47 (12): 53–55. http://dl.acm.org/citation.cfm?id=1035165.

Bogost, Ian, and Nick Montfort. 2009. "Platform Studies: Frequently Questioned Answers." In *Digital Arts and Culture*. http://escholarship.org/uc/item/01r0k9br.pdf.

Brin, Sergey, and Larry Page. 1998. "The Anatomy of a Large-Scale Hypertextual Web Search Engine." *Computer Networks and ISDN Systems* 30 (1): 107–117. http://www.sciencedirect.com/science/article/pii/S016975529800110X.

Bucher, Taina. 2012. "Programmed Sociality: A Software Studies Perspective on Social Networking Sites". Oslo, Norway: University of Oslo. http://tainabucher.com/wp-content/uploads/2009/08/Bucher_Ph.D.diss_.pdf.

Chhabra, Sidharth, Anupama Aggarwal, Fabricio Benevenuto, and Pponnurangam Kumaraguru. 2011. "Phi. Sh/$ oCiaL: The Phishing Landscape through Short URLs." In *Proceedings of the 8th Annual Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference*, 92–101. http://dl.acm.org/citation.cfm?id=2030387.

Cutts, Matt, and Jason Shellen. 2005. "Preventing Comment Spam." *Official Google Blog*. January 19. http://googleblog.blogspot.nl/2005/01/preventing-comment-spam.html.

De Maeyer, Juliette. 2011. "How to Make Sense of Hyperlinks? An Overview of Link Studies." In *A Decade in Internet Time: Symposium on the Dynamics of Internet and Society*. http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1926301.

———. 2013. "Towards a Hyperlinked Society: A Critical Review of Link Studies." *New Media & Society* 15 (5) (August 1): 737–751. doi:10.1177/1461444812462851. http://nms.sagepub.com/content/15/5/737.

Digg. 2006. "Digg External Story Submission Process." http://videos.revision3.com/diggnation/digg_submission_062506.pdf.

Elmer, Greg, and Ganaele Langlois. 2013. "Networked Campaigns: Traffic Tags and Cross Platform Analysis on the Web." *Information Polity* 18 (1) (January 1): 43–56. doi:10.3233/IP-2011-0244. http://dx.doi.org/10.3233/IP-2011-0244.

Finkelstein, Seth. 2008. “Google, Links, and Popularity versus Authority.” In *The Hyperlinked Society*, edited by Joseph Turow and Lokman Tsui, 104–124. Ann Arbor, MI: The University of Michigan Press. http://www.webuse.org/p/c09/.

Foot, K., S. M. Schneider, M. Dougherty, M. Xenos, and E. Larsen. 2003. “Analyzing Linking Practices: Candidate Sites in the 2002 US Electoral Web Sphere.” *Journal of Computer-Mediated Communication* 8 (4): 0–0. http://onlinelibrary.wiley.com/doi/10.1111/j.1083-6101.2003.tb00220.x/full.

Garrett, Sean. 2010. “Links and Twitter: Length Shouldn’t Matter.” *Twitter Blog*. http://blog.twitter.com/2010/06/links-and-twitter-length-shouldnt.html.

Gehl, Robert W. 2010. “A Cultural and Political Economy of Web 2.0”. Fairfax, VA: George Mason University. http://digilib.gmu.edu/dspace/handle/1920/5886.

Gerlitz, Carolin, and Anne Helmond. 2013. “The Like Economy: Social Buttons and the Data-Intensive Web.” *New Media & Society* 15 (8) (November 20): 1348–1365. doi:10.1177/1461444812472322. http://nms.sagepub.com/content/15/8/1348.abstract.

Giles, Jim. 2011. “The Web’s Crystal Ball Gets an Upgrade.” *MIT Technology Review*. November 14. http://www.technologyreview.com/news/426105/the-webs-crystal-ball-gets-an-upgrade/.

Gillespie, Tarleton. forthcoming. “The Relevance of Algorithms.” In *Media Technologies*, edited by Tarleton Gillespie, Pablo Boczkowski, and Kirsten Foot. Cambridge, MA: MIT Press. http://www.tarletongillespie.org/essays/Gillespie%20-%20The%20Relevance%20of%20Algorithms.pdf.

———. 2010. “The Politics of ‘platforms.’” *New Media & Society* 12 (3): 347–364. http://nms.sagepub.com/content/12/3/347.short.

Halavais, Alexander. 2008. “The Hyperlink as Organizing Principle.” In *The Hyperlinked Society*, edited by Joseph Turow and Lokman Tsui, 39–55. Ann Arbor, MI: The University of Michigan Press. http://www.webuse.org/p/c09/.

Hargittai, Eszter. 2008. “The Role of Expertise in Navigating Links of Influence.” In *The Hyperlinked Society*, edited by Joseph Turow and Lokman Tsui, 85–103. Ann Arbor, MI: The University of Michigan Press. http://www.webuse.org/p/c09/.

Hourihan, Meg. 2002. “What We’re Doing When We Blog.” *O’Reilly Network* (June 13). http://www.oreillynet.com/pub/a/javascript/2002/06/13/megnut.html.

Hsu, Chien-leng, and Han Woo Park. 2011. “Sociology of Hyperlink Networks of Web 1.0, Web 2.0, and Twitter: A Case Study of South Korea.” *Social Science Computer Review* 29 (3): 354–368. http://ssc.sagepub.com/content/29/3/354.short.

John, Nicholas A. 2012. “Sharing and Web 2.0: The Emergence of a Keyword.” *New Media & Society*. http://nms.sagepub.com/content/early/2012/06/29/1461444812450684.abstract.

Johnson, Bobbie, Charles Arthur, and Josh Halliday. 2010. “Libyan Domain Shutdown No Threat, Insists Bit.ly.” *The Guardian*, October 9, sec. Technology. http://www.guardian.co.uk/technology/2010/oct/08/bitly-libya.

Kirkpatrick, Marshall. 2009. “Twitter Crowns Bit.ly As The King of Short Links; Here’s What It Means.” *ReadWrite*. May 6. http://readwrite.com/2009/05/06/twitter_crowns_bitly_as_the_king_of_short_links_he.

Langlois, Ganaele, Greg Elmer, Fenwick McKelvey, and Zachary Devereaux. 2009. “Networked Publics: The Double Articulation of Code and Politics on Facebook.”

Canadian Journal of Communication 34 (3). http://www.cjc-online.ca/index.php/journal/article/viewArticle/2114.

Langlois, Ganaele, Fenwick McKelvey, Greg Elmer, and Kenneth Werbin. 2009. “Mapping Commercial Web 2.0 Worlds: Towards a New Critical Ontogenesis.” *Fibreculture* (14). http://www.journal.fibreculture.org/issue14/issue14_langlois_et_al_print.html.

Lee, Sangho, and Jong Kim. 2013. “WarningBird: A Near Real-Time Detection System for Suspicious URLs in Twitter Stream.” *IEEE Transactions on Dependable and Secure Computing* 10 (3): 183–195. doi:10.1109/TDSC.2013.3.

Maggi, Federico, Alessandro Frossi, Stefano Zanero, Gianluca Stringhini, Brett Stone-Gross, Christopher Kruegel, and Giovanni Vigna. 2013. “Two Years of Short URLs Internet Measurement: Security Threats and Countermeasures.” In *Proceedings of the 22nd International Conference on World Wide Web*, 861–872. WWW ’13. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee. http://dl.acm.org/citation.cfm?id=2488388.2488463.

Manovich, Lev. 2013. *Software Takes Command*. New York, NY: Bloomsbury Academic. http://www.bloomsbury.com/us/software-takes-command-9781623567453/.

Mattlemay. 2011. “Bitly and Verisign.” *Bitly Blog*. http://blog.bitly.com/post/12296694165/bitly-and-verisign.

Megiddo, Nimrod, and Kevin S. McCurley. 2005. “Efficient Retrieval of Uniform Resource Locators.” http://www.google.com/patents/US6957224.

Neumann, Alexander, Johannes Barnickel, and Ulrike Meyer. 2010. “Security and Privacy Implications of Url Shortening Services.” In *Proceedings of the Workshop on Web 2.0 Security and Privacy*. http://w2spconf.com/2011/papers/w2sp-urlShortening-slides.pdf.

Niederer, Sabine, and José Van Dijck. 2010. “Wisdom of the Crowd or Technicity of Content? Wikipedia as a Sociotechnical System.” *New Media & Society* 12 (8) (December 1): 1368–1387. doi:10.1177/1461444810365297. http://nms.sagepub.com/content/12/8/1368.

Nisenholtz, Martin. 2008. “The Hyperlinked News Organization.” In *The Hyperlinked Society*, edited by Joseph Turow and Lokman Tsui, 104–124. Ann Arbor, MI: The University of Michigan Press. http://www.webuse.org/p/c09/.

O’Reilly, Tim. 2007. “What Is Web 2.0: Design Patterns and Business Models for the next Generation of Software.” *Communications & Strategies* (1): 17–37. http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1008839.

Park, Han Woo. 2003. “Hyperlink Network Analysis: A New Method for the Study of Social Structure on the Web.” *Connections* 25: 49–61.

Puschmann, Cornelius, and Jean Burgess. 2013. “The Politics of Twitter Data.” In *HIIG Discussion Paper Series No. 2013-01*. http://papers.ssrn.com/abstract=2206225.

Rieder, Bernhard. 2013. “Scrutinizing a Network of Likes on Facebook (and Some Thoughts on Network Analysis and Visualization).” *The Politics of Systems*. http://thepoliticsofsystems.net/2013/07/scrutinizing-a-network-of-likes-on-facebook-and-some-thoughts-on-network-analysis-and-visualization/.

Rogers, Richard. 2002. “Operating Issue Networks on the Web.” *Science as Culture* 11 (2): 191–213. http://www.tandfonline.com/doi/abs/10.1080/09505430220137243.

———. 2006. *Information Politics on the Web*. Cambridge, MA: The MIT Press.

———. 2013. *Digital Methods*. Cambridge, MA: MIT Press.

Rogers, Richard, and Anat Ben-David. 2008. "The Palestinian-Israeli Peace Process and Transnational Issue Networks: The Complicated Place of the Israeli NGO." *New Media & Society* 10 (3) (June 1): 497–528. doi:10.1177/1461444807085321. http://nms.sagepub.com/content/10/3/497.

Rose, Kevin. 2006. "Integrating Digg Within Your Website." *Digg Blog*. http://diggtheblog.blogspot.nl/2006/07/integrating-digg-within-your-website.html.

Ruppert, Evelyn, John Law, and Mike Savage. 2013. "Reassembling Social Science Methods: The Challenge of Digital Devices." *Theory, Culture & Society* 30 (4) (July 1): 22–46. doi:10.1177/0263276413484941. http://tcs.sagepub.com/content/30/4/22.

Schachter, Joshua. 2009. "On Url Shorteners." *Joshua Schachter's Blog*. http://joshua.schachter.org/2009/04/on-url-shorteners.html.

Singletary, Taylor. 2011. "Twitter's T.co URL Wrapper Is Now on for All URLs 19 Characters and Greater." *Twitter Developers Discussions*. https://dev.twitter.com/discussions/1062.

Sotomayor, Bernardo Rafael. 1998. "Automatic Summary Page Creation and Hyperlink Generation." http://www.google.com/patents/US5708825.

Sterling, Greg. 2009. "URL Shorteners Come To Google & Facebook." *Search Engine Land*. December 14. http://searchengineland.com/url-shorteners-come-to-google-facebook-31880.

Turow, Joseph. 2008. "Introduction: On Not Taking the Hyperlink for Granted." In *The Hyperlinked Society: Questioning Connections in the Digital Age*, edited by Joseph Turow and Lokman Tsui. Ann Arbor: University of Michigan Press.

Twitter. 2011. "Link Sharing Made Simple." *Twitter Blog*. https://blog.twitter.com/2011/link-sharing-made-simple.

———. 2012. "Discover: Improved Personalization Algorithms and Real-Time Indexing." *Twitter Blog*. May 1. https://blog.twitter.com/2012/discover-improved-personalization-algorithms-and-real-time-indexing.

"Twitter Privacy Policy." 2012. May 17. https://twitter.com/privacy.

Ullrich, Carsten, Kerstin Borau, Heng Luo, Xiaohong Tan, Liping Shen, and Ruimin Shen. 2008. "Why Web 2.0 Is Good for Learning and for Research: Principles and Prototypes." In *Proceedings of the 17th International Conference on World Wide Web*, 705–714. WWW '08. New York, NY, USA: ACM. doi:10.1145/1367497.1367593. http://doi.acm.org/10.1145/1367497.1367593.

"URL Shorteners Make the Web Substantially Slower; Facebooks' Fb.me Is Slowest." 2010. *The Official WatchMouse Blog*. http://blog.watchmouse.com/2010/03/url-shorteners-make-the-web-substantially-slower-facebooks-fb-me-is-slowest/.

Van Dijck, José. 2013. "Facebook and the Imperative of Sharing." In *The Culture of Connectivity: A Critical History of Social Media*. New York, NY: Oxford University Press.

W3C. "Links." http://www.w3.org/TR/html401/struct/links.html.

Walker, Jill. 2002. "Links and Power: The Political Economy of Linking on the Web." In *Proceedings of the Thirteenth ACM Conference on Hypertext and Hypermedia*, 72–73. http://dl.acm.org/citation.cfm?id=513358.

Weltevrede, Esther. 2011. “Discussion during the Digital Methods Initiative Winter Conference 2011” University of Amsterdam.

Winer, Dave. 2009. “Solving the TinyUrl Centralization Problem.” *Scripting News*. http://scripting.com/stories/2009/03/07/solvingTheTinyurlCentraliz.html.

Appendix 1

URL FOLLOW with cURL

INPUT:

```
http://t.co/THHo3lRM
```

OUTPUT:

```
HTTP/1.1 301 Moved Permanently

cache-control: private,max-age=300

date: Sat, 22 Jun 2013 11:16:28 GMT

expires: Sat, 22 Jun 2013 11:21:28 GMT

location: http://fb.me/1pNxOnFcT

server: tfe

Content-Length: 0


HTTP/1.1 301 Moved Permanently Location:

http://www.huffingtonpost.com/2012/01/14/costa-concordia-disaster-_n_1206167.html

Content-Type: text/html; charset=utf-8

X-FB-Debug: rVCZC2NA//jyoJbrKkd7cPzOppS/20yueM0IvLpBwts=

Date: Sat, 22 Jun 2013 11:16:28 GMT

Connection: keep-alive

Content-Length: 0
```

```
HTTP/1.1 200 OK
Server: Apache
P3P: CP='NO P3P'
Content-Type: text/html; charset=utf-8
Date: Sat, 22 Jun 2013 11:16:28 GMT
Connection: keep-alive
```