Int. J. Human-Computer Studies 96 (2016) 38–53

Contents lists available at ScienceDirect

Int. J. Human-Computer Studies

journal homepage: www.elsevier.com/locate/ijhcs

Short links and tiny keyboards: A systematic exploration of design trade-offs in link shortening services

Sandy J.J. Gould [a,*], Anna L. Cox [a], Duncan P. Brumby [a], Sarah Wiseman [b]

[a] UCL Interaction Centre, University College London, 66-72 Gower Street, WC1E 6BT, United Kingdom
[b] Department of Computing, Goldsmiths, University of London, London SE14 6NW, United Kingdom

ARTICLE INFO

Article history:
Received 7 March 2016
Received in revised form
23 July 2016
Accepted 25 July 2016
Available online 28 July 2016

Keywords:
Link shortening
Short links
Keyboard
Optimization
Typing
Data entry
Mobile
Touchscreen

ABSTRACT

Link-shortening services save space and make the manual entry of URLs less onerous. Short links are often included on printed materials so that people using mobile devices can quickly enter URLs. Although mobile transcription is a common use-case, link-shortening services generate output that is poorly suited to entry on mobile devices: links often contain numbers and capital letters that require time consuming mode switches on touch screen keyboards. With the aid of computational modeling, we identified problems with the output of a link-shortening service, *bit.ly*. Based on the results of this modeling, we hypothesized that longer links that are optimized for input on mobile keyboards would improve link entry speeds compared to shorter links that required keyboard mode switches. We conducted a human performance study that confirmed this hypothesis. Finally, we applied our method to a selection of different non-word mobile data-entry tasks. This work illustrates the need for service design to fit the constraints of the devices people use to consume services.

1. Introduction

Semantically meaningful URLs are often long. This makes them tedious to transcribe. QR codes obviate the need to type, but are not human-readable and have a number of issues with usability (Shin et al., 2012) and security (Vidas et al., 2013). Link shortening services like *bit.ly* provide a compromise: the process of typing complex and lengthy URLs is accelerated and human readability is preserved. Unfortunately, many shortening services exhibit little consideration for how links might be made quick and easy to type. The outputs of shortening services usually contain a mix of numbers and mixed-case letters. On space-constrained mobile devices, entering these characters requires changing the keyboard from lowercase mode to number mode or uppercase mode. Making these mode switches to access different characters is particularly time consuming and error prone (Greene et al., 2014). Shorter links, therefore, may not necessarily be faster to type.

Generating short links that do not require keyboard mode switches necessarily means using a smaller selection of characters. For a given link length this means fewer unique links can be generated. This reduction in the number of options for each character of a link can be mitigated by increasing the length of links. But how much longer would such a link need to be? In this paper we model the process of text entry on three widely-used mobile platforms. We use simulations to systematically explore the output of a popular shortening service, *bit.ly*. To see whether the predictions we make based on our model hold true in reality we test them in a game-like human performance study.

We found that link shorteners trade-off link length and entry difficulty. Most shorteners optimize too aggressively for link length; their output is awkward to transcribe. We show that links can be made easier to type with only a modest increase in their length. Given the increasingly limited functional utility of ultra-short links on services like Twitter, link shortening services should prioritize making links easier to type.

1.1. Related work

Efforts to make text entry easier for people have mostly focused on improving entry interfaces, like keyboards (e.g., Cheng et al., 2013; Leiva et al., 2015; Oulasvirta et al., 2013). Another way to improve text entry is to adapt input interfaces to accommodate the kinds of input that they are most likely to receive. Wiseman et al. (2013) showed that in hospitals the distribution of digits entered into devices is not random. Input interfaces that are designed for a particular set of possible inputs perform better than standard interfaces that do not take account of the strings that are likely to be entered (Wiseman et al., 2013). The preponderance of third-party

* Corresponding author.
E-mail address: s.gould@cs.ucl.ac.uk (S.J.J. Gould).

http://dx.doi.org/10.1016/j.ijhcs.2016.07.009

mobile keyboards designed specifically for entering emoji characters or calendar events also reflects this kind of thinking: make the interface fit the input.

Unfortunately, implementing bespoke data-entry interfaces is a luxury that is rarely available to designers. People do not have custom keyboards installed for every input use case. More often than not, the design of input interfaces is entirely out of the control of service designers. In most circumstances input values should instead fit interfaces (see, e.g., Wiseman et al., 2016).

The entry of certain types of input can be suboptimal when keyboard designs have had to make trade-offs. This is especially the case for keyboards on phone-sized touchscreen devices. To save space, one of the concessions the designers of touchscreen keyboards make is to only show lowercase characters. Accessing uppercase letters and numbers requires multiple taps to change a keyboard's mode.

Gallagher and Byrne (2015) modeled the effect of having to switch modes on touchscreen keyboards in the context of password entry. They found that the interval between typing two lowercase letters was around 500 ms. The interval between typing a lowercase letter and an uppercase letter was approximately 1500 ms, a three-fold increase. Mode switching on small touchscreen keyboards is costly, but implementing a custom keyboard solely for entering passwords is impractical: touchscreens don't have the space to display the full set of numbers, letters and special characters in a single keyboard pane.

If input interfaces are fixed constraints in a system, we should consider how a service might be adjusted so as to better fit those constraints in likely contexts of use. This is not always possible, but there are scenarios in which target information can be substituted or altered without compromising a service. We focus on a particular example that exhibits this property: link shortening services.

1.2. Link shortening services

Link shortening services like *bit.ly*, *ow.ly* or *goo.gl* act as intermediaries between users and websites. Users provide a target link, for instance, https://www.elsevier.com/journals/international-journal-of-human-computer-studies/1071-5819/guide-for-authors. A shortening service generates a much shorter link, in this case, http://bit.ly/1OT4BPc. A mapping between these long and short links is stored by the shortening service. When a short URL is requested users are redirected to the original long URL.

Shorter links come at a cost: semantic information is lost from a URL. In the example above, a user reading the long link has a good idea that they'll end up on an Elsevier page. The short link, however, might just lead to Rick Astley's *Never Gonna Give You Up* (http://bit.ly/e4Rt5rr). A user would not know until they had followed it. The loss of semantic information has made shortening services a vector for phishing attacks (Chhabra et al., 2011; Klien and Strohmaier, 2012). Despite these shortcomings, shortening services have a number of benefits.

Short links are useful when space is limited or when characters are restricted. Shortening services also offer social media users methods for tracking engagement. Short links are often easier to copy and paste because they are more compact and generally do not cover multiple lines. Short links can also ease transcription from physical artefacts to digital devices. Short links feature on print advertising (see Fig. 1) and on slides during talks. By using shortened links, labyrinthine directions to a slide deck on a university server can be shortened to a few quick keystrokes.

1.3. Improving link shorteners

How well do existing link shortening services meet the requirements of the use cases we have discussed so far? RFC 3986 (Berners-Lee et al., 2005), which defines how URLs work, specifically discusses the tension between the digital and physical use of URLs: "[URL] design considerations," it says, "are not always in alignment". Given that trade-offs are required, do shortening services make reasonable ones?

One of the constraints on link shortening schemas is the need to be able to address a large potential set of links so that each short link can be guaranteed to be unique. Shorteners like *bit.ly* produce seven-character identifiers comprising numbers and mixed-case characters. In this scheme, each character can be any of 62 options: 26 uppercase letters, 26 lowercase letters or 10 digits. This yields a large space of possible identifiers, $(26+26+10)^7$, or around 3.5 trillion.

In the digital domain, unique identifiers could be made much shorter by allowing more than 62 options for each character. Many modern browsers (but not all services) support *percent encoded* URLs. UTF-8, a method of encoding Unicode characters, supports up to 1,112,064 characters. Using the full array of UTF-8 characters, a link shortening schema could exceed the size of the *bit.ly* pool of possible identifiers by orders of magnitude using only three characters (i.e., $1{,}112{,}064^3 \gg 62^7$). This yields links that are physically smaller and use fewer characters. Additionally, a link like *bit.ly/→∴♣* is no more or less meaningful than a link like *bit.ly/E5tF68G*. In the digital domain, where links are clicked and their composition after shortening is immaterial, shorteners could increase their effective compression ratio by using expanded

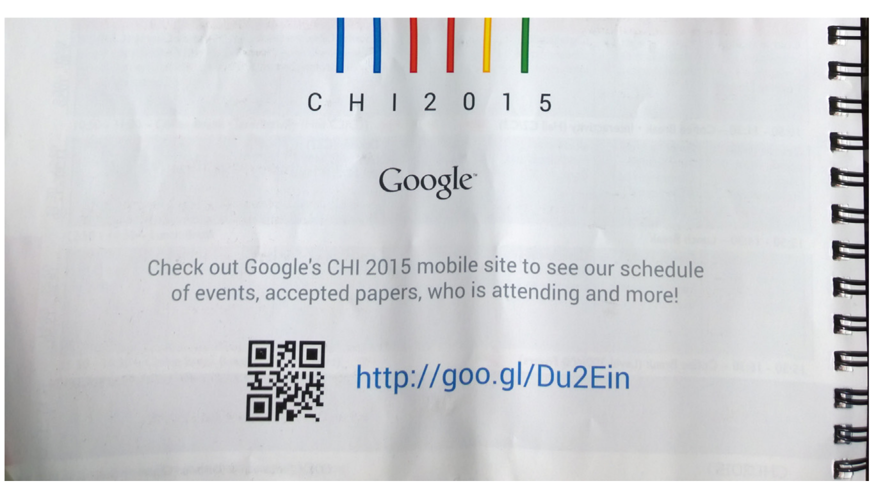

Fig. 1. Short links often appear on print advertising. Here the link contains numbers, capitals and lowercase characters.