

A Natural Language URL-Shortener

David Branner
Hack and Tell
eBay, New York
20140610

URL shorteners and their benefits are well known –

URL shorteners and their benefits are well known –
create very short pointer to a long URL:

URL shorteners and their benefits are well known –

create very short pointer to a long URL:

<http://bit.ly/TGtQtH>

URL shorteners and their benefits are well known –

create very short pointer to a long URL:

<http://bit.ly/TGtQtH>

➔ *<https://www.hackerschool.com/manual#sec-history>*

URL shorteners and their benefits are well known –

create very short pointer to a long URL:

<http://bit.ly/TGtQtH>

➔ *<https://www.hackerschool.com/manual#sec-history>*

less susceptible to corruption in transmission

URL shorteners and their benefits are well known –

create very short pointer to a long URL:

<http://bit.ly/TGtQtH>

➔ *<https://www.hackerschool.com/manual#sec-history>*

less susceptible to corruption in transmission
concealment of final destination

key element: variable part of a shortened URL (“path”)

key element: variable part of a shortened URL (“path”)

- <http://7.ly/iMau>
- <http://2.gp/zkSE>
- <http://qr.net/Bozx>
- <http://bit.ly/TGIQtH>
- <http://x.vu/KC4s4e>
- <http://ow.ly/xQNbx>
- <https://t.co/ZfUR4euiph>

key element: variable part of a shortened URL (“path”)

- <http://7.ly/iMau>
- <http://2.gp/zkSE>
- <http://qr.net/Bozx>
- <http://bit.ly/TGIQtH>
- <http://x.vu/KC4s4e>
- <http://ow.ly/xQNbx>
- <https://t.co/ZfUR4euiph>

ugglesome in the extreme

key element: variable part of a shortened URL (“path”)

- <http://7.ly/iMau>
- <http://2.gp/zkSE>
- <http://qr.net/Bozx>
- <http://bit.ly/TGIQtH>
- <http://x.vu/KC4s4e>
- <http://ow.ly/xQNbx>
- <https://t.co/ZfUR4euiph>

ugglesome in the extreme
compressed, hence rarely readable

key element: variable part of a shortened URL (“path”)

- <http://7.ly/iMau>
- <http://2.gp/zkSE>
- <http://qr.net/Bozx>
- <http://bit.ly/TGIQtH>
- <http://x.vu/KC4s4e>
- <http://ow.ly/xQNbx>
- <https://t.co/ZfUR4euiph>

ugglesome in the extreme
compressed, hence rarely readable
easy to remember?

**“custom” shortened URLs can be easier to remember
because they rely on natural language ability**

**“custom” shortened URLs can be easier to remember
because they rely on natural language ability**

<http://bit.ly/HStory>

**“custom” shortened URLs can be easier to remember
because they rely on natural language ability**

`http://bit.ly/HStory`

➔ `https://www.hackerschool.com/manual#sec-history`

**“custom” shortened URLs can be easier to remember
because they rely on natural language ability**

`http://bit.ly/HStory`

➔ `https://www.hackerschool.com/manual#sec-history`

easier to remember

**“custom” shortened URLs can be easier to remember
because they rely on natural language ability**

`http://bit.ly/HStory`

➔ `https://www.hackerschool.com/manual#sec-history`

easier to remember

short only if you get there first (tend to be long)

Why not have URL pointers that are always
both readable and very short?

Why not have URL pointers that are always
both readable and very short?

(Talon-sharpening exercise at Hacker School recently.)

Why not have URL pointers that are always
both readable *and* very short?

(Talon-sharpening exercise at Hacker School recently.)

The trick to picking always-readable short URL-paths:

Why not have URL pointers that are always
both readable and very short?

(Talon-sharpening exercise at Hacker School recently.)

The trick to picking always-readable short URL-paths:

use the characters for the most common Chinese words

Chinese characters

Chinese characters

2635 in the current official HSK proficiency exam

Chinese characters

2635 in the current official HSK proficiency exam

2635 \in simplified; 2671 \in traditional;

1692 $\{x : x \in \{\text{simp} \cap \text{trad}\} \}$

Chinese characters

2635 in the current official HSK proficiency exam

2635 \in simplified; 2671 \in traditional;

1692 $\{x : x \in \{\text{simp} \cap \text{trad}\} \}$

compare to paired Roman letters (of either case):

$52 \times 52 = 2704$

Chinese characters

2635 in the current official HSK proficiency exam

2635 \in simplified; 2671 \in traditional;

1692 $\{x : x \in \{\text{simp} \cap \text{trad}\} \}$

compare to paired Roman letters (of either case):

$$52 \times 52 = 2704$$

kind	random 字	random Roman digraph
#	2635	2704

Chinese characters

2635 in the current official HSK proficiency exam

2635 \in simplified; 2671 \in traditional;

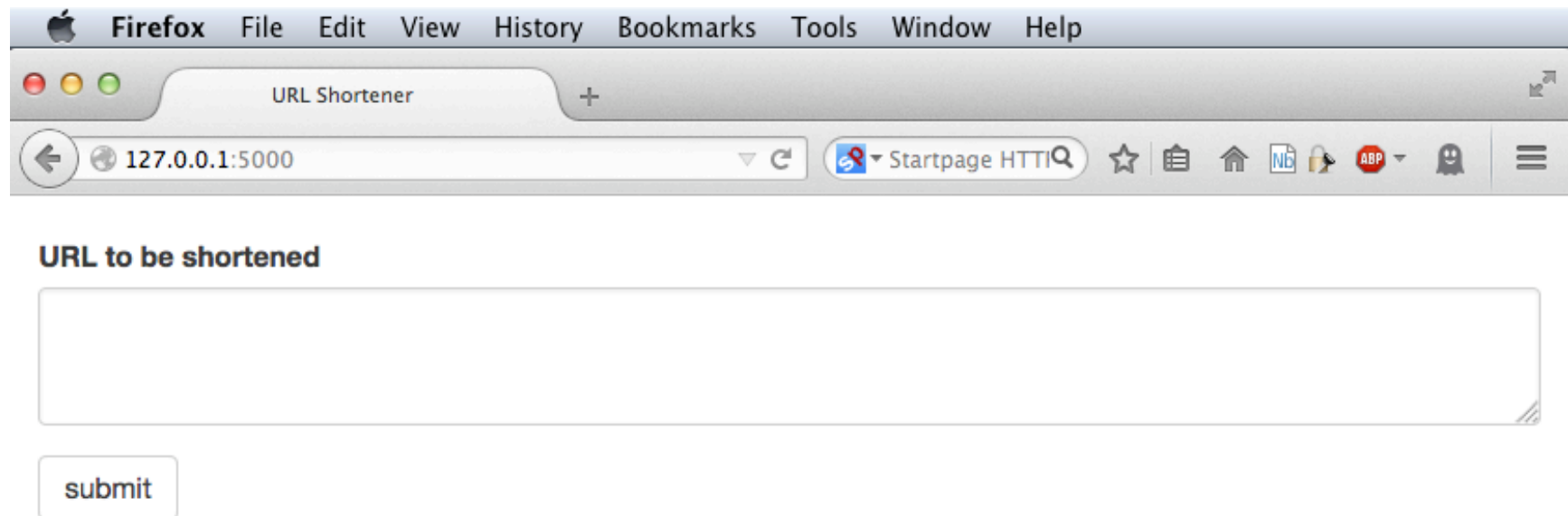
1692 $\{x : x \in \{\text{simp} \cap \text{trad}\} \}$

compare to paired Roman letters (of either case):

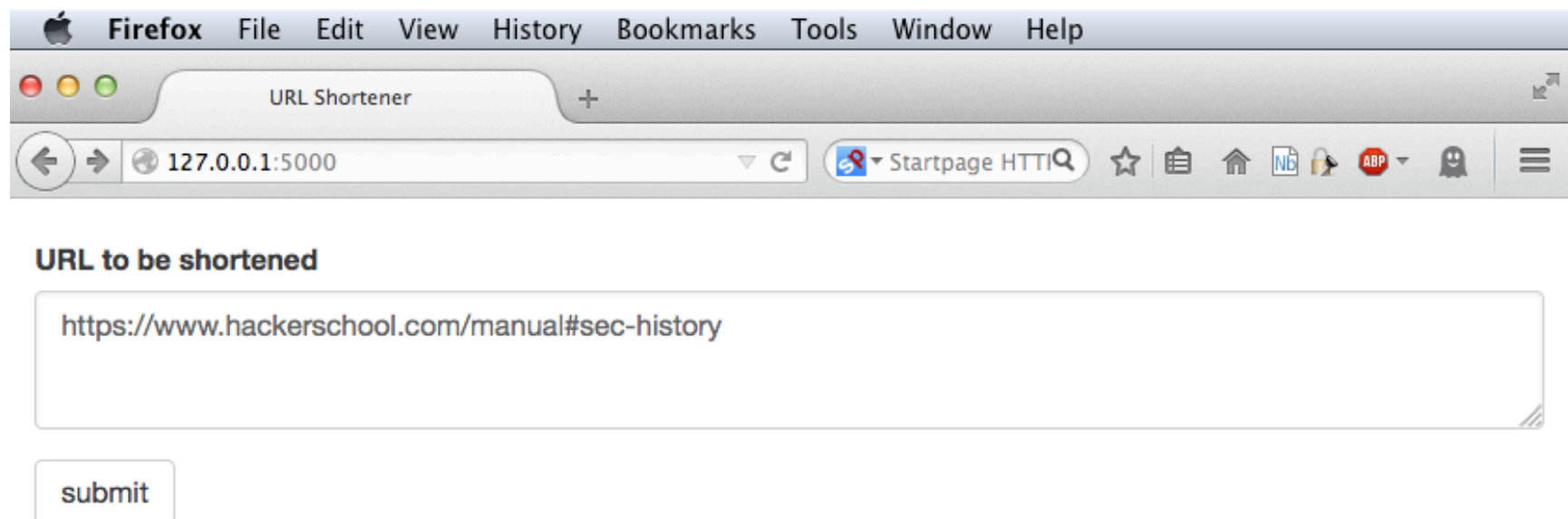
$$52 \times 52 = 2704$$

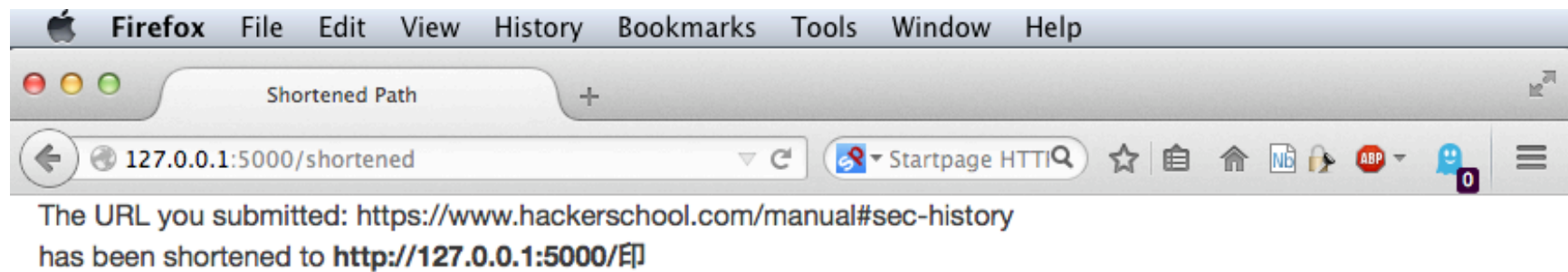
kind	random 字	random Roman digraph
#	2635	2704
readable?	guaranteed	probably not

Code for proof-of-concept on my public Git repository.



*Branner, A Natural Language URL-Shortener
Hack and Tell, 20140610. p. 28/40*





First c. 2650 URLs ➔ one character

`http://127.0.0.1:5000/印`

First c. 2650 URLs → one character

`http://127.0.0.1:5000/印`

Next c. $2650 \times 2650 = 7022500$ URLs → two characters

`http://127.0.0.1:5000/厉吉`

First c. 2650 URLs → one character

`http://127.0.0.1:5000/印`

Next c. $2650 \times 2650 = 7022500$ URLs → two characters

`http://127.0.0.1:5000/厉吉`

Next 18,609,625,000 URLs → three characters

`http://127.0.0.1:5000/天鼻歪`

Always readable

Always readable (may not make sense...)

Note:

Always readable (may not make sense...)

Note: many custom shorteners allow Chinese characters:

<http://bit.ly/史>

Always readable (may not make sense...)

Note: many custom shorteners allow Chinese characters:

<http://bit.ly/史>

Also note:

Always readable (may not make sense...)

Note: many custom shorteners allow Chinese characters:

<http://bit.ly/史>

Also note: the advantage of shortening to Chinese does not mean bandwidth savings:

Always readable (may not make sense...)

Note: many custom shorteners allow Chinese characters:

`http://bit.ly/史`

Also note: the advantage of shortening to Chinese does not mean bandwidth savings:

`http://bit.ly/史` may be sent from your browser as

`http://bit.ly/%E5%8F%B2`

劇
終