Analyzing the Keystroke Dynamics of Web Identifiers

Andrew G. West
Verisign Labs, Verisign Inc.
Reston, VA, USA
awest@verisign.com

ABSTRACT

Web identifiers such as usernames, hashtags, and domain names serve important roles in online navigation, communication, and community building. Therefore the entities that choose such names must ensure that end-users are able to quickly and accurately enter them in applications. Uniqueness requirements, a desire for short strings, and an absence of delimiters often constrain this name selection process.

To gain perspective on the speed and correctness of name entry, we crowdsource the typing of 51,000+ web identifiers. Surface level analysis reveals, for example, that typing speed is generally a linear function of identifier length. Examining keystroke dynamics at finer granularity proves more interesting. First, we identify features predictive of typing time/accuracy, finding: (1) the commonality of character bi-grams inside a name, and (2) the degree of ambiguity when tokenizing a name – to be most indicative. A machine-learning model built over 10 such features exhibits moderate predictive capability. Second, we evaluate our hypothesis that users subconsciously insert pauses in their typing cadence where text delimiters (*e.g.,* spaces) would exist, if permitted. The data generally supports this claim, suggesting its application alongside algorithmic tokenization methods, and possibly in name suggestion frameworks.

CCS CONCEPTS

• **Human-centered computing** → *Interaction devices*; *Keyboards*; • **Information systems** → *Personalization*; Online advertising; • **Social and professional topics** → Internet governance / domain names;

KEYWORDS

typeability; keystroke dynamics; keyboards; typos; web identifier; domain names; hashtags; usernames

WebSci '17, June 25-28, 2017, Troy, NY, USA

ACM ISBN 978-1-4503-4896-6/17/06…$15.00
https://doi.org/10.1145/3091478.3091482

1 INTRODUCTION

Little needs to be said about the pervasiveness of web identifiers, with 325+ million domain names [29] enabling web navigation/email and billions of social network accounts. The need for careful identifier selection in these naming systems is evidenced both empirically and anecdotally.

For domain names, consider that [16] estimates that *typosquatting* names – registrations intended to capture traffic from misspellings of popular websites – received an estimated 68.2 million clicks/day in 2010. While most such domains are monetized via ad revenue, roughly 4% are leveraged for "hit stealing" and competitive advantage [1]. Proactive name selection could minimize this threat vector. Even in the absence of such threats, consider one registrar's leading tip for domain name selection: "make it easy to type" [22].

Parallel motivations have been seen in other naming systems. For example, "brandjacking" on social networks and less nefariously, the digital land rush when Facebook enabled usernames in 2009 [19]. Mature namespaces demand nuanced identifier selection and evaluation criteria, to include typeability metrics. Our research will reveal a user's ability to "parse" names absent delimiters as an important property, for which Twitter provides an instructive example. Hashtag `#nowthatchersdead`, intended to be parsed as `#now|thatchers|dead` following the death of former UK prime minister Margaret Thatcher, was instead met by grief for the still living American celebrity Cher, being interpreted as `#now|that|chers|dead` [14].

To learn about how end users enter web identifiers we crowdsource 51,002 domain name typings (Sec. 2). We focus on domains in particular because of the: (1) ubiquity of the DNS, (2) availability of related metadata (*e.g.,* popularity), and (3) ability to experiment with a hierarchical namespace. Our experiment masquerades as a CAPTCHA usability study to deflect from our true measurements, wherein the minimally obfuscated CAPTCHA text is generated from domain names. Given the crowdsourced nature of our data, we are attentive to the potential biases of a self selected and pay-per-task workforce.

We then analyze the millisecond-granularity timing data, answering research questions along three themes (Sec. 3):

(1) *Broad measurements*: How does identifier length affect typing time? What is the relationship between backspace usage and typing time?

(2) *Features/models predicting typeability*: Can the keyboard mapping or lexical properties of a name predict input speed and/or correctness? What is the accuracy of a model leveraging all such features?

(3) *Tokenization clues in typing latency*: Do users provide hints about where word boundaries lie with irregularities in their typing cadence?

umclimbing	norwegianblues	monicaandbryce
oaklandbakery	makehomenow	monserrattreviews
loveishypnotic	charliegunn	lokalsportlive
philsmithroofing	laguardiatux	trianglehall
letsbuytickets	killnoise	goldmedalplumbing

Table 1: Example SLDs from the "random" subset

It is from an exploration of these, and related topics, that we identify the key contributions of this work:

- Linear growth in typing times as identifier length increases (at ≈290 ms/char.) reveal no hidden costs in seeking out longer names in mature namespaces
- Familiarity brings increased typeability for both domains and their hierarchical extensions ("random" identifiers are typed 16% slower than "popular" ones)
- Features capturing how an identifier maps onto a physical keyboard (*e.g.,* finger travel distance) are relatively poor predictors of typeability
- Lexical properties are more useful; domains are easier to type when they: (1) can be easily tokenized into component words, and (2) are composed of character bi-grams common in English text
- Models to predict typing times have tepid effectiveness with RMSE=0.31 where the regression target is normalized about 1.0. For a 12-character domain, one standard deviation of confidence spans 2.8 seconds.
- The maximum intra-character latency for an identifier occurs on a word boundary 3× to 4× more often than random chance, yielding helpful tokenization clues

These findings can be integrated into identifier suggestion tools, or serve as guidance to the individuals, corporations, and investors who participate in these naming systems.

To the best of our knowledge, this is the first research to measure the typeability of web identifiers, although it builds on literature about typeability and keystroke dynamics. These related works are reviewed (Sec. 4) before noting future work and concluding (Sec. 5).

2 DATA COLLECTION

To produce keystroke measurements one must first decide what strings will be typed (Sec. 2.1). These are at the core of our experiment design and its crowdsourced distribution (Sec. 2.2). After collecting 51k measurements, we take a cursory look at the data, then scrutinize the workforce and the biases it may introduce (Sec. 2.3).

2.1 Identifier Selection

Domain names are representative of the broader class of web identifiers; sharing their core characteristics, while their name structure and usage data enable additional analysis. This structure is seen in a domain like `example.com`, where "com" is the *top-level domain* (TLD) or *extension* and "example" is the *second-level domain* (SLD). This work does not consider depths beyond the second-level, so we also use the terms *left-of-dot* and *right-of-dot* to refer to the SLD and TLD, respectively. Our primary focus is on SLDs, where the most creative free will is exercised (Sec. 2.1.1). We treat SLDs and TLDs independently, with the ability to comparatively measure TLD typeability a side effect of using domain strings (Sec. 2.1.2).

TLD	P%	GROUPING
COM	20%	
NET	10%	Legacy TLDs (40%)
ORG	10%	
DE	10%	
CN	10%	CC TLDs (30%)
UK	10%	
XYZ	10%	
CLUB	10%	New gTLDs (30%)
ONLINE	10%	

Table 2: TLD selection probabilities (P%)

2.1.1 Second-level Domain Strings. It is our goal to collect typing measurements for SLDs that are: (1) consistent with those typed during ordinary Internet usage, and/or (2) consistent with the class of names an end-user concerned with typeability might consider for registration. To this end, we create two domains sets:

(1) POPULAR: Alexa's top 10,000 domains
(2) RANDOM: Subset of the COM/NET zone where domains that a crawler flagged as "redirecting", "parked", or "pay-per-click advertising" are discarded

Next, we have Amazon Mechanical Turk (MTurk) workers *tokenize* or *segment* a subset of these domains into their component words (*e.g.,* `somedomain.com` → `some|domain`). Workers were told to visit the websites in making this determination, with a 23.6 second average task completion time suggesting these instructions were heeded. Skipping subtleties in the interest of brevity, a minimum of 3 workers and a maximum of 8 workers are asked to tokenize a name until consensus was reached (75%+ of workers agreeing on a tokenization) or the domain was abandoned. These authoritative human tokenizations are critical to Sec. 3.3.

The final dataset contains 14,650 "random" and 7,250 "popular" names. After stripping off the TLD[1] these 21,900 SLDs form the basis of our experimental set. Tab. 1 shows example SLDs drawn from the "random" subset.

2.1.2 Domain Extensions. Retaining the TLDs associated with our ≈22k SLDs would lead to an overwhelming majority of COM and NET names. For purposes of experimenting with TLD typeability, we instead randomly apply extensions to SLD labels per the probabilities of Tab. 2. While this gives rise to some names which are not registered, we assume the "dot" offers a consistent break in typing cadence such that the SLD and TLD may be measured independently. Tab. 2 also shows the selected TLDs draw from three different groupings:

[1]The public suffix list (`http://publicsuffix.org`) is used to determine "effective TLDs". For instance, "co.uk" is the effective TLD for `example.co.uk`.

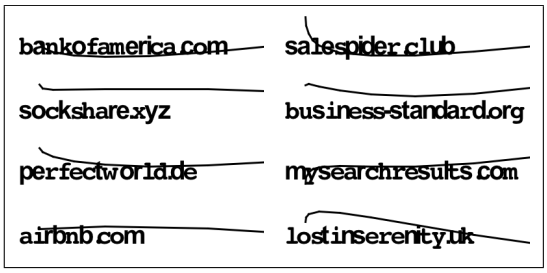

Figure 1: Example domain CAPTCHA images, shown here at reduced size

(1) *Legacy TLDs* were delegated in 1985 and have pervasiveness that could manifest as typing "muscle memory". We test the 3 (`COM`; `NET`; `ORG`) most popular [29].

(2) *Country code extensions* (ccTLDs) are two-letter TLDs that likely have geographically variable familiarity. We have chosen the 3 (`CN`; `DE`; `UK`) most popular per [29], after ommitting the special case of `TK`.

(3) *New generic TLDs* (new gTLDs) were first deployed in 2013 and have expanded the available extension space. Our selection (`XYZ`; `CLUB`; `ONLINE`) opts for the most popular TLDs marketed towards English-speaking markets at the time of our early 2016 analysis [26].

2.2 Experiment Design

Our 21,900 domains are *what* will be typed. Next is *how* the typing data is collected. We opt to generate domain text as CAPTCHA images (Sec. 2.2.1). Then, we have MTurk workers participate in a purported CAPTCHA usability study, where we instrument the input forms to collect keyboard data (Sec. 2.2.2). In this manner we hope to avoid the Hawthorne effect: telling a participant we are observing typing behaviors potentially modifies such behaviors.

This crowdsourced CAPTCHA strategy does have drawbacks, as discussed subsequently. However, we find it preferable to alternatives such as in-person laboratory experiments that would have high cost and lack scalability. Creating a browser plug-in to monitor native interactions with the URL bar might produce the best data, but would face immense privacy and adoption challenges.

2.2.1 CAPTCHA Generation. For each of our domains (SLDs from Sec. 2.1.1 with TLDs randomly appended per Tab. 2) we generate a CAPTCHA image using the SimpleCaptcha library [24]. Fig. 1 displays some example images. All text is generated in lowercase, and a minimal amount of obfuscation is applied. Our goal is to keep the domain text very legible, adding just enough noise to make our CAPTCHA storyline plausible.

2.2.2 MTurk Instrumentation. These CAPTCHA images are central to the MTurk experiments. Fig. 2 shows the worker interface, coded in HTML, Javascript, PHP, with a MySQL backend. The following points, as annotated in Fig. 2, highlight design decisions:

❶ *Instruction set*: Instructions are straightforward in explaining our CAPTCHA usability study. A bullet makes explicit that all answers are of the form "[text].[text]" after initial trials found frequent dot omissions.

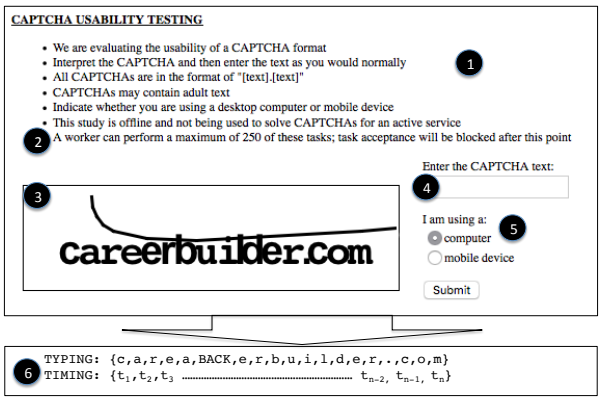

Figure 2: MTurk worker interface

❷ *Participation limit*: No worker can complete more than 250 tasks, limiting one to contributing 0.5% of the dataset. This balances the need to find sufficient workers with the desire to capture typing behaviors of a broad population. The limit is enforced in software.

❸ *CAPTCHA image*: As generated in the previous section, we randomly select from available images (associated with a puzzleID) when the page is generated.

❹ *Solution form*: Where the worker types the CAPTCHA text. The form is instrumented using the JQuery `keydown()` event, which was found to have the most consistent cross-browser behavior. Both the key pressed and the millisecond timing of the event are recorded.

❺ *Device type*: Workers are asked whether they are using a desktop or mobile device. Differing keyboards have obvious ramifications on typing measurements.

❻ *Output*: After "submit" is pressed, we store every key entered and the associated timings. We also have the generated puzzleID, the User Agent string of the worker, and MTurk identifiers/metadata.

The complexity of our interface is such that it cannot be managed via MTurk's web application. We utilize Amazon's command-line tools [3] for task launching and management.

2.3 Data Quality & Workforce

Over two weeks in early 2016, 60,500 typing experiments were launched paying $0.02 each. Well-formed data came back for 60,135 (99.39%) of these with the few errors assumed to be from atypical browser configurations. We now sanity check this data (Sec. 2.3.1) before examining the workforce that produced it (Sec. 2.3.2).

2.3.1 Data Scrubbing. In a cursory pass of that data, we discard typings that are either (1) incorrect or (2) from a mobile device, a filtering that is visualized in Fig. 6.

A typing is considered *incorrect* if replaying the sequence of characters keyed does not produce a result which matches the generating text of the CAPTCHA displayed. We emphasize that "incorrect" does not preclude the use of backspace. The dataset has 51,491 (85.6%) correctly keyed entries and 8,644 (14.4%) incorrect ones, with a number of circumstances giving rise to these faults:

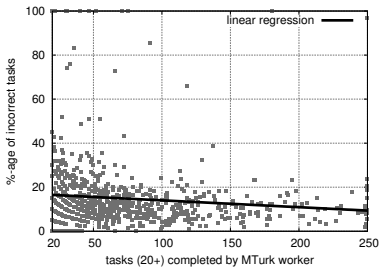

Figure 3: Worker task quantity × incorrect typings[2]

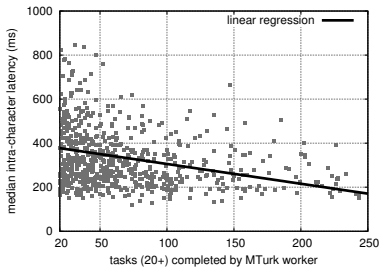

Figure 4: Worker task quantity × typing latency[2]

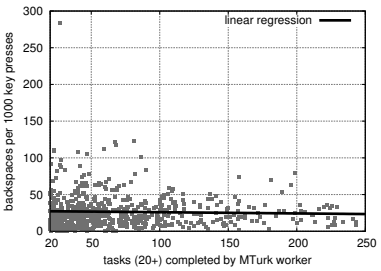

Figure 5: Worker task quantity × backspace usage[2]

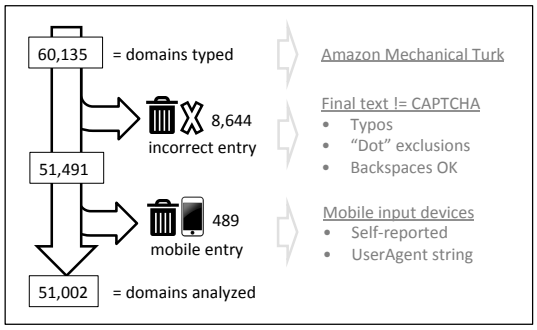

Figure 6: Discarding incorrect and mobile typings

Figure 7: CDF of task distribution over workers

- *Mouse usage*: Mouse actions such as highlighting characters or moving the cursor are manipulations not captured in keyboard data. It is unknown how many "incorrect" typings are the result of such effects.
- *No solution text*: 416 (4.8%) incorrect tasks have blank solution fields. Heavy overlap with mobile device usage suggests these are users with Javascript disabled.
- *Omitted periods*: Despite explicit instructions about generated CAPTCHA structure, 1941 non-blank incorrect typings (22.5%) contained no dot character.
- *Character errors*: The remaining mistakes are typographical or transcription errors. Transpositions, missing characters, duplicate characters, and missing plurality were common.

These 8,644 incorrect typings are discarded. The "character errors" set may be useful in the study of typos but is of less utility in our statistical context here, *i.e.,* if a worker keys more characters than was in the CAPTCHA text, how does one treat the length property?

The second condition for discarding a task is if it was *typed using a mobile device*. There were 1,246 self-reported mobile inputs, only 477 of which were entered correctly, presumbly due to Javascript issues. Analyzing user agent strings confirmed these and found a further 12 cases indicating a mobile browser. Discarding these 489 typings should not be interpreted as irrelevance of mobile typing behaviors. Instead, they demand study with crowdsourcing and instrumentation better tuned for mobile environments.

The 51,002 remaining domain/CAPTCHA typings are the primary basis for analysis in the remainder of this work.

2.3.2 Workforce & Bias. We now examine the MTurk workers who contributed to our typing experiments, with a particular eye towards the fact they are a compensated and self selected workforce:

Basic statistics: Over ≈65k tasks there were 1442 unique workers, completing at median 21 tasks each, with the average being 47. The maximum of 250 tasks was reached by 20 workers. Fig. 7 plots the task CDF, showing the top 20% of workers did 50% of all work, and the top 50% of workers accounted for 90% of tasks. Such a distribution is not unexpected, as workers often sample tasks before committing to considerable work. The experiments were popular with batches of 5000 tasks being completed in under an hour, implying our payout was competitive.

Paid workforce: Workers average around 6 seconds/task, which if sustained is $6/hour after Amazon commission. This is non-trivial for workers in developing economies, thus incentivizing them to work quickly through a finite and dissipating task pool. A possible side effect is making "incorrect" typings that we discard, with the portion that are correct being entered at atypically high speeds. Counteracting this is the notion of reputation on MTurk and the ability to halt payouts to underperforming workers. Fig. 3 shows that our most prolific workers commit about 7 less errors per 100 tasks than those who contributed 20 typings.[2] We do not believe this is a severe affect and attribute it primarily to familiarity with interface and CAPTCHA formats.

[2]These figures do not plot workers who performed <20 tasks. They contribute only 10% of tasks and lack a body of work of sufficient size and consistency from which to draw conclusions.

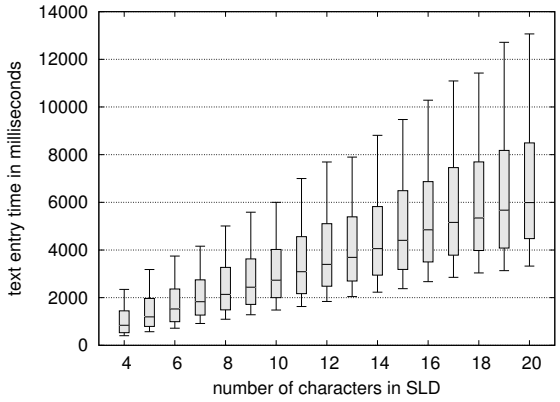

Figure 8: Distribution of SLD typing times, by length[3]

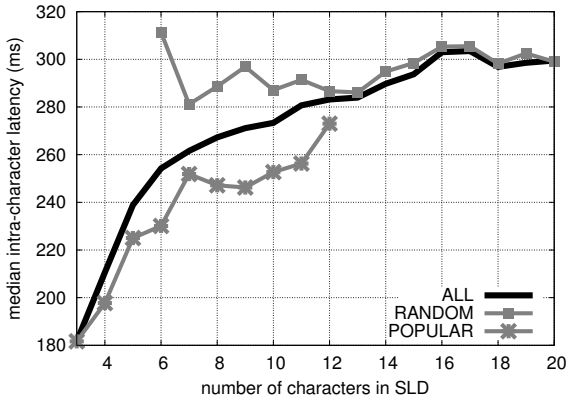

Figure 9: Median intra-char. latency by SLD length. Data points backed by less than 500 typings are not plotted.

Self-selected workers: Workers on the MTurk platform choose the work they perform. Therefore, if one finds a task aligns poorly with their skill set (*e.g.,* typing speed), they might stop performing it to seek more lucrative alternative work. This leaves a non-representative labor pool to perform the bulk of the tasks. Fig. 4 plots the *median intra-character latency* (MICL) by worker, one metric summarizing typing speed. The linear regression shows an MICL of ≈400ms for workers who did 20 tasks, but workers who did 250 tasks were nearly twice as quick at ≈200ms. This is a rather profound difference. It is partially because familiarity breeds speed: For workers who did 150+ tasks, their first 25 tasks were typed at a rate 7.55% slower than subsequent ones. Still, this cannot account for much of the gap. We remain mindful of this bias while noting this work is most concerned with relative, not absolute measures of keystroke dynamics.

It is interesting to note, per Fig. 5, that backspace usage is roughly constant across task quantity. Combined with Fig. 4, this implies that susceptibility to errors/backspace does not correlate strongly with typing speed.

Non-measurable effects: Other biases do not lend themselves to quantification. For example, most MTurk workers are from India or the US [11]. We also make the assumption that typing behaviors when entering CAPTCHA solutions are consistent with those when interacting with web identifiers in a URL bar or email client.

3 DATA ANALYSIS

The dataset of 51k domain typings is now analyzed towards three major goals: First, measuring the speed and correctness of domain typings at both SLD and TLD levels (Sec. 3.1). Second, developing features and producing models predictive of typing times (Sec. 3.2). Third, assessing the feasibility of predicting word boundaries using intra-character latency (Sec. 3.3).

3.1 Broad Measurements

In order to assess SLD and TLD typing speed and correctness, one must determine what keystrokes define these components. When domains are entered without using backspace, this is straightforward. Let's assume correctly typing domain `foo.com` produces a timing vector T of length $|T| = n = 7$: $T = \{t_1, t2, t_3, t_4, t_5, t_6, t_7\}$. We then define a function, $T_{di} = dot\_index(T) = t_4$, that determines where the "dot" character lies in the typing vector. With this we can subset T into $T_{SLD} = \{t_1, t_2 \dots T_{di-1}\}$ and $T_{TLD} = \{T_{di+1}, T_{di+2} \dots t_n\}$ with typing times $time(T_{SLD}) = (T_{di-1} - t_1)$ and $time(T_{TLD}) = (t_n - T_{di+1})$, respectively.

Backspace characters introduce complications. For example, take the typing sequence:

…which is a correct typing of `foo.com`. Now, $dot\_index(\cdot)$ is extended to return only the index of the dot character that persists in the final output (the second dot, in the above example). The dataset has 713 examples of a dot character being erased; 3% of all backspace usages. With this established, we can now do analysis at SLD (Sec. 3.1.1) and TLD granularity (Sec. 3.1.2).

3.1.1 Second-level Domain Strings. Users have the most creative free will in choosing second-level domains. Our analysis emphasizes length ramifications:

Typing speed by length: Fig. 8 shows SLD typing time as broken down by SLD length[3]. The increasing typing times are equal parts insight and sanity check. We find 4 character SLDs are typed in about a second, with 20 character domains taking 6 seconds at median. The 91st percentile of workers types about twice as slow as median typists.

Impact of increasing lengths: Focusing only on the median time growth as length increases, Fig. 9 examines if typing speed is a linear function of SLD length. As namespaces mature, short names become less available, and it is crucial to understand if there is any penalty (beyond linear) of choosing longer, more descriptive names.

The graph shows there are incremental penalties for "popular" names. Indeed, the Alexa top 10k exhibits a positive correlation between popularity and length, *i.e.,* the most trafficked domains

[3]For all box plots, the box markings correspond to the typical 1Q, median, and Q3 of the distribution. The whiskers mark the 9th and 91st percentiles. Workers who pause mid-task sometimes return typing times measured in minutes, one reason we prefer to characterize timing metrics at median.

	COM	NET	DE	ORG	UK	ONLINE	CLUB	XYZ	CN
domains (doms)	10,412	5,248	4,971	5,387	5,171	4,989	5,030	4,783	5,011
length (len)	3	3	2	3	2	6	4	3	2
chars. (c = doms × len)	31,326	15,744	9,942	16,161	10,342	24,945	20,120	14,349	10,022
backspaces typed (bs)	295	254	205	363	291	725	592	752	619
chars. per backspace (c/bs)	105.88	61.98	48.50	44.52	35.54	34.41	33.99	19.08	16.19
TLD per backspace ((c/bs)/len)	35.29	20.66	24.25	14.84	17.77	6.88	8.50	6.36	8.10

Table 3: Backspace statistics broken out by TLD[4]

PROPERTY	RAND	POP	ALL
# SLDs	33,403	17,599	51,002
SLDs w/backspace	6,014	2,215	8,229
minimal keypresses	462,036	148,224	610,260
actual keypresses	489,048	158,380	647,428
backspace (bs) presses	14,018	5,266	19,284
avg. SLD length (chars.)	13.8	8.4	12.0
% SLDs w/bs	18.0%	12.6%	16.1%
actual/min keypresses	105.9%	106.9%	106.1%
bs/actual presses	2.9%	3.3%	3.0%

Table 4: Backspace statistics for SLDs

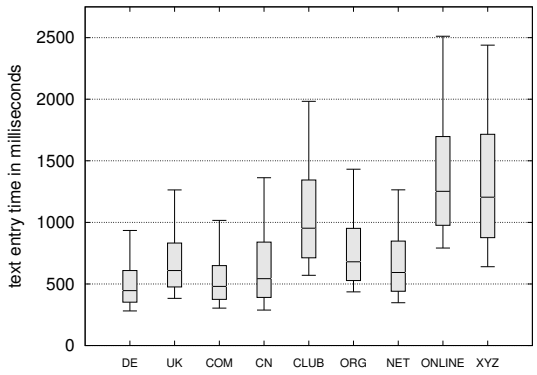

Figure 10: Distribution of TLD typing times[3,4]

tend to be shorter. Consider that popular sites are financially well-positioned to acquire very short names, such as Facebook's $8.5M buy of `fb.com`. Thus, it may be the case that the short names very high in the Alexa rankings are simply being typed atypically fast.

Regardless, popularity is primarily a byproduct of site content, not name selection. All new registrations enter the market as "random" ones so it is more instructive to look at that subset for guidance. There we see a more linear pattern of around 290ms being added for each additional character. We believe for new registrants and the name suggestion task that this is the more salient statistic.

Backspace usage: Tab. 4 displays statistics concerning backspaces in SLD typing. Highlights are that 16% of all domains typed utilized the backspace character, with 3% of all keypresses being backspace. Breaking out the "random" and "popular" sets seemingly reveal gaps with just 12.6% of popular names having a backspace compared to 18% of random ones. However, after the 5 character average length difference is normalized, ratios (*e.g.,* actual/min keypresses) show the error rate per character is quite similar. More informally, because random TLDs are longer they give typists more opportunity to err. The fact that error rate is relatively constant even when a typist is (presumably) familiar with a name hints (see Sec. 3.2) that predictors of typographical errors may tough to identify.

3.1.2 Top-level Domain Strings. Given finite TLD options, we are able to compare relative typeability across a sample of extensions[4]:

Typing speed: Fig. 10 visualizes the typing time distribution by TLD. We see legacy extensions [29] are typed very quickly, even comparably to the two-character ccTLDs. New gTLDs tend to be typed slower than the other classes, perhaps a reflection of the fact only 52% of consumers are aware of any of the new extensions [10]. Once normalized by length the per-character cost of a TLD is ≈250ms, though `COM` and `NET` are outliers that tend to be typed much quicker.

Backspace usage: Tab. 3 presents TLD backspace usage in both raw and length normalized form. Consistent with the timing statistics, `COM` fares favorably, *e.g.,* only once per 35 typings of "com" do we expect someone to make a mistake; every 106 characters.

3.2 Features/Models Predicting Typeability

While retroactive insights regarding domain text entry are interesting, mining that data enables models which can be applied to unregistered identifiers. Such models could be applied in name suggestion and marketing tools. We first develop features indicative of typeability (Sec. 3.2.1) before evaluating their performance individually and in combination (Sec. 3.2.2). Our primary goal is predicting typing speed for a provided string, although we briefly explain challenges in trying to predict backspace usage (Sec. 3.2.3).

3.2.1 Feature Development. Features are developed in 2 themes: (1) keyboard topology and (2) string properties. We describe each of our 10 features and provide intuition on how they might be predictive. Tab. 5 provides a succinct feature reference.

Keyboard topology: We hypothesize a string's mapping onto the physical keyboard layout offers typeability clues. Though statistics are scarce, we assume universal QWERTY keyboard usage across our primarily Indian and U.S.-based workforce [11]. We repurpose a keyboard measurement library [5] to help calculate these features:

- `distance`: Distance (in meters) traveled by all fingers in typing a string, including moving from, and if necessary, returning to, the home row. Assumes 1.8cm square keys and a standard QWERTY layout; a popular measure in evaluating keyboard efficiency.

[4]Recalling Sec. 2.1.2, this TLD selection compares the most popular extensions of each type per industry statistics [26, 29].

FEATURE	RRIF	DESCRIPTION
distance	0.04	finger travel distance (meters)
same_hand_%	0.10	%-age key/trans. on same hand
same_fing_%	0.04	%-age key/trans. on same finger
row1_%	0.16	%-age key/trans. to/from numeric keys
repeat_%	0.04	%-age key/trans. back to same char.
pinky_%	0.10	%-age keys typed with pinky
dom_length	0.05	length of sld in characters
seg_words	0.06	number of words in SLD; per corpus
seg_diff	0.49	ease of tokenizing SLD; per algorithm
2gram_prob	0.16	log-sum of bigram probabilties in SLD

Table 5: Features for typing time prediction. Regressional ReliefF (RRIF) [21] ranks feature efficacy. RRIf*100 is given here.

- same_hand_%: Percentage of key transitions that use the same typing hand. For example, a [q] followed by a [b] might be an awkward stretch.
- same_fing_%: Percentage of key transitions that utilize the same typing finger. Since the finger will not start resting on a home row character, some typists might incur latency in returning home.
- row1_%: Percentage of key transitions that go from the alphabetical keys to the number/hyphen ones and vice-versa. The top row of keys require the longest reach, and we have observed typists abandoning touch typing to find/reach them.
- repeat_%: Percentage of key transitions that repeat the same character, on the presumption these can be typed very quickly since no finger travel is required.
- pinky_%: Percentage of key transitions using the pinky finger. Tangential evidence [7] suggests the little finger is least accurate in striking characters.

String properties: The remaining 4 features are based on lexical characteristics of the string itself:

- dom_length: Domain length is an obvious feature if trying to predict actual typing times, but more subtle if doing regression around length normalized values.
- seg_words: Recall from Sec. 2.1.1 that all SLDs have a consensus tokenization from human workers. Thus it is known how many words compose an SLD, our feature here. Given our hypothesis in Sec. 3.3 that word boundaries cause pauses in typing cadence, word quantity should impact typing time.
- seg_diff: We extend the straightforward algorithmic tokenization method of [13] with numerical and hyphen support. This method considers all 2^{n-1} possible tokenizations of a length n string and sorts them by a relative "fitness" score based on English word commonality [17] with a preference for longer words. The ratio of the highest to the second-highest tokenization score is our feature here, proxying how ambiguous the tokenization appears on the surface (recall workers actually visited the webpages). We imagine quick identification of component keywords makes domain entry more like normal typing, with familiar letter sequences.
- 2gram_prob: Related work suggests that "familiar" sequences are easier to type [31]. Fig. 11 plots the English bi-gram (*i.e.,* two character) probabilities from a corpus of English text [17] against the intra-character latency for those bigrams per our experiments. Regression reinforces that claim. Accordingly, we encode a feature:

$$\frac{\Sigma\ |log_{10}(bigram\_prob)|\ \forall\ bigrams \in SLD}{length(SLD) - 1}$$

which sums bigram commonality in a string, uses *log* to put very tiny probabilities back in a reasonable number space, and normalizes by string length.

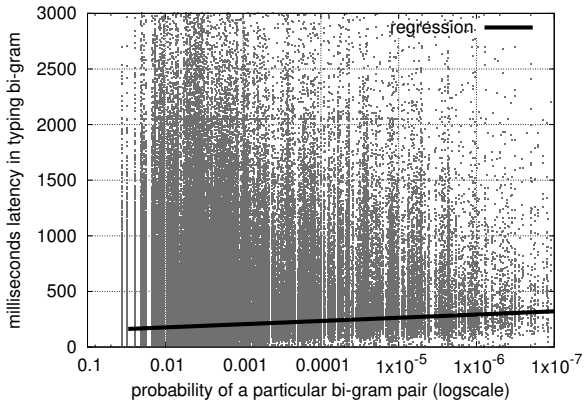

Figure 11: Commonality of bi-gram pairs in English text × typing speed in our experiments

3.2.2 Models & Performance. We next address the prediction target, individual feature performance, model construction, and overall performance:

Predicted values: Given a string, predicting an absolute typing time is difficult. As earlier graphs showed, typing speed between individuals can vary greatly. Instead, we want a dependent variable (*i.e.,* target) that is typist agnostic. This is achieved by first considering only workers with a sufficient body of work: 20+ tasks. The median intra-character latency (MICL) for each worker is calculated. Then, for every SLD he/she has typed[5], one can divide the actual timing time by the expected typing time:

$$\frac{actual\_typing\_time}{(length(SLD) - 1) * MICL_{worker}}$$

to yield a ratio on $(0, \infty)$. These ratios are our regression target, one we believe to be an intuitive representation of typing speed. A ratio < 1.0 indicates a quickly typed string, while ratios greater than 1.0 suggest a slower and more difficult one. While the median ratio is 1.0, the long tail stretches towards slower typing times given their unbounded nature. In practice, few values exist outside of $[0.4, 2.2]$.

Feature performance in isolation: Returning to Tab. 5, rankings show that most keyboard topology features exhibit middling to low performance. Such metrics are often leveraged to assess alternative keyboard layouts, and the 12 character average SLD length

[5]For training, testing, and worker MICL we use exclusively SLDs from the "random" set as these more closely approximate the available names that would be evaluated by potential registrants.

probably provides insufficient opportunity for patterns to emerge. The `row1_%` feature is the exception. Although only 14.3% of SLDs combine numeric/hypen and alphabetical characters, the penalty is very large where it can be leveraged.

Lexical features show most promise, with features `seg_diff` and `2gram_prob` being highest ranked. Simply put, domains that have straightforward tokenizations and do not contain rare bigrams will be quickly typed. The dominant weight of `seg_diff` in particular suggests typeability may be improved by choosing longer, keyword-rich strings rather than introducing abbreviations or ambiguity into shorter identifiers.

Model construction & evaluation: To go from features to a predictive model the Weka [8] implementation of support vector regression (SVR) [23] is utilized. Though the actual SVR model is more complex and accurate, we present one built using linear regression over normalized feature values on [0,1] in the interest of human interpretation:

```
+0.7254
+0.6537 * (norm) row1_%
+0.5336 * (norm) 2gram_prob
-0.3262 * (norm) seg_diff
+0.1214 * (norm) distance
+0.0825 * (norm) dom_length
+0.0781 * (norm) pinky_%
+0.0752 * (norm) same_fing_%
+0.0222 * (norm) same_hand_%
```

One thing to notice is that 9 of 10 features are additive in nature. That is, we have identified what a typeable domain should *not* contain if one desires quick entry. For evaluation purposes, 10-fold cross validation is used.

Model performance: The best model parameterization produces a correlation coefficient of 0.2043, mean absolute error of 0.2417, and a root mean squared error (RMSE) of 0.3132. Focusing on RMSE, let's imagine a 12 character string is provided to our model. Suppose the model returns 1.0, predicting the actual typing time will match the expected typing time. Based on Fig. 8, we can estimate that 4 seconds is a reasonable expected typing time for a length 12 SLD. Going ±RMSE in each direction: $(1 - 0.35 = 0.65 * 4$ seconds) = 2.6 seconds and $(1 + 0.35 = 1.35 * 4$ seconds) = 5.4 seconds, one can state the typing time for the provided string will be between 2.6 and 5.4 seconds with one standard deviation ($\approx$ 70%) of confidence.

In no way do we claim this to be a strong result. Indeed, for the majority of predictions which will cluster about 1.0, this broad confidence interval will encompass typing times ranging from rather quick to moderately slow. Given this, we believe the model has most utility in flagging strings with acute typeability concerns (*e.g.,* scores > 1.5). Because so few features significantly contribute, it would also be easy to communicate corrective actions to a user.

3.2.3 Backspace Prediction. As with typing speed, there are use-cases for algorithms that can predict typing errors. To some extent the prior model already captures this, since backspace usage manifests as additional typing time. However, a purpose built model could better focus on susceptibility to typosquatting and indicate where in a string mistakes might occur. Recall that true typos were discarded in Sec. 2.3.1, leaving only examples with corrected errors. However, it stands to reason that circumstances giving rise to backspace usage are similar to those of persistent typos.

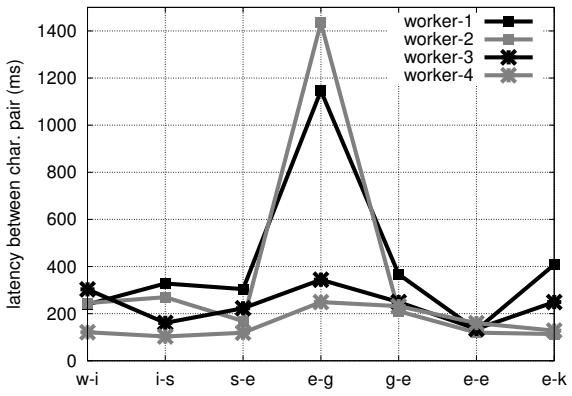

Figure 12: Intra-character latencies for a SLD (`wisegeek`)

Preliminary to *where* a backspace will occur is *if* one will occur in a string. This binary formulation lends itself to classification algorithms such as random forests, but unfortunately reusing the feature set of Tab. 5 produces a model that outperforms random chance by only a few percentage points. This begs questions about the ineffectiveness of the model and the tractability of the problem.

Perhaps error patterns are unique to typists and not shared amongst broader populations. Having a maximum of ≈3,000 characters per worker, we are not positioned to explore this hypothesis. Also consider that whereas time elapses *between* keypresses, errors occur *on* keypresses, so they might be less context dependent. Plotting the most mis-keyed characters normalized by occurrence reveals that $\{a, e, i, o\}$ are least error prone and $\{-, c, v, g\}$ are most errror prone. However, all of these differences contribute nominally to the classification accuracy. Gaining little traction, we leave progress on the this task as future work.

3.3 Tokenization Clues in Typing Latency

Statistical methods are the norm when tokenizing web identifiers; recall the word fitness technique based on [13] that was applied in Sec. 3.2.1. When that technique is run over the ≈28k domains for which we have consensus human tokenizations (per Sec. 2.1.1), the accuracy is roughly 75%. Given short lengths and the intersection of many natural languages it is easy to see why the task is a difficult one. Sec. 4 will describe how some tokenization methods use contextual data (*e.g.,* the domain content or associated Twitter message) to improve performance. However, such clues are not available for unregistered identifiers. This motivated a search for alternative tokenization strategies.

Arriving at keystroke dynamics, we hypothesize that users leave small pauses in their typing cadence in locations where they believe word boundaries to exist. Fig. 12 shows a deliberately selected example of how such a pattern might manifest. There, 4 different workers have typed the `wisegeek` SLD, which is tokenized as `wise|geek`, as a surface-level parsing would suggest. For each of the 4 workers, albeit to varying degrees, the longest intra-character latency lies on the word boundary. We seek to understand how common this is

across the typing dataset. We begin with a proof-of-concept experiment that isolates the hypothesis under very favorable conditions. Then, these constraints are relaxed to learn about potential of the method, before discussing how it might be useful in practice.

Sanity checking: In order to test for typing latency at word boundaries, we identify a set of domain typings which are atypically well-suited for the task. The properties of such a SLD/typing are:

- The name should contain no numbers or hyphens, given "row 1" characters independently incur high latency
- The MTurk workers who came to a consensus tokenization must be in 100% agreement, and that human tokenization should have exactly two words
- Algorithmic tokenization methods should produce the same result as the human workers
- The ratio between the first and second fittest algorithmic tokenizations (our earlier `seg_diff` feature) should not be in the 1Q of that distribution
- Eligible typings should contain no backspaces

These criteria leave 10,034 typings for analysis. If one were to consider domains of more than two words, one would need to identify local maxima as possible word boundaries. Our simpler setup allows us to predict the maximum intra-character latency as the sole tokenization point. The average length amongst these 10k identifiers is 10.97 characters. This implies there are 9.97 possible positions, on average, where the word boundary might lie, giving one a 10.03% chance of randomly guessing the correct boundary.

Results show 41.6% of the time the maximum latency is "correct" in corresponding to the boundary, a 4× lift over random. The average length of correct instances is 10.65 characters, similar enough to the 10.97 overall average to conclude we are not just lucky on short names. Further, 63.7% of the time the word boundary falls on the character pair with the highest or 2nd-highest latency. While not definitive, it is clear these latencies offer tokenization clues.

Practical evaluation & application: The prior experiments were unfairly skewed in favor of our hypothesis. Indeed, the cases where the algorithmic tokenization is confident in its result are precisely those where one does *not* need to rely on keystroke dynamic clues.

There are 2,864 typings that meet the criteria above, except that that segmentation fitness ratio falls in the lowest 1Q; the technique being unsure in its tokenization. In these cases, the maximum latency correctly identifies the word boundary 31.5% of the time (3× lift). Another interesting set are the 3,127 typings where the algorithmic tokenization result and the authoritative one provided by workers disagree. Here, 34.0% of the time the boundary is correctly identified, as per agreement with the human segmentation.

Though not accurate enough for standalone tokenization, these results show latency measurements have utility, particularly where algorithmic methods show ambiguity. It is unclear how well this applies to name suggestion, where the creative process of typing/exploring such names may differ from our straightforward experiments. Still, other applications can be imagined. For instance, latency timings could indicate one intended to type a space character when instead two words were accidentally concatenated, perhaps in/on a mobile device or search engine form field.

4 RELATED WORK

To the best of our knowledge, this is the first research to measure the typeability of web identifiers. Although the potential of such measurements has been proposed [28], we are the first to explore the topic empirically. Indeed, rarely does one get to choose *what* will be typed. Instead, the bulk of typeability research concentrates on the efficacy of different keyboard layouts or devices (*e.g.,* mobile, eye tracking) over longer blocks of text.

However, there is an earlier body of work – some of it predating the mainstream Internet – that examines the relationship between speech, handwriting, and typing behaviors. For example, Zesiger *et al.* [31] contribute that "actual words" are typed quicker than "pseudo words", and that word commonality positively correlates with quicker entry speeds, mirroring findings from the handwriting domain. Priva [18] draws a similar parallel between typing times and the speed/ease of speaking a given word, in addition to surveying many studies showing that "typing is sensitive to language based effects." Gentner's work [9] motivates our feature development and latency hypothesis. First, he shows that typing latency is relatively fixed between sequences of characters, but cannot be determined from characters in isolation. Second, he speaks about the role of word boundaries in resetting typing cadence.

Though "keystroke dynamics" broadly refers to any usage of detailed keyboard timing data, more recently is has become almost synonymous with its application as a biometric marker and authentication mechanism. Proposed in [15], the idea is that individuals have typing signatures that differentiate them from most other typists. Although a miscreant may have stolen your password, they are unlikely to type it in the same manner as you would, raising a security flag. Banerjee and Woodard [4] survey progress in this space, with [12] standardizing a typing corpus and taking a comparative approach. Our work is distinct in that it characterizes the behavior of the entire population of typists rather than trying to identify distinctions between them. As such, our work is similar to [6] which uses keystroke dynamics to develop classifiers for 15 different emotional states. Though they were able to capture native typing behaviors using keylogging, their study recruited just 12 participants, highlighting the challenges of such collections.

This work has utilized existing statistical tokenization algorithms and contributed a novel non-statistical method. Regarding the former, our preference has been the straightforward approach of [13]. Other methods exist, including those espousing multiple dictionary corpora [25], sequential character streams [2], and contextual information from webpage structure [30]. With these, domains and hashtags are the primary strings of interest [20, 25]. Our corpus of 28k crowdsourced domain tokenizations could prove useful to this community, as the 3k expert-sourced segmentations of Srinivasan *et al.* [25] are the only comparable data of which we are aware.

Finally, even our preliminary attempts at predicting typos and where they might occur faltered. While excellent economic [16, 27] and longitudinal [1] studies on typosquatting have been conducted, they answer an easier question: "given a popular domain X and a lexically similar domain Y, is Y typosquatting?". Moore and Edelman [16] introduce "fat finger distance", a metric capturing that mis-struck keys tend to be spatially adjacent to the intended characters. Szurdi *et al.* [27] leverage WHOIS, DNS, Alexa rankings,

and HTML content atop lexical clues in making typosquatting determinations. While both well capture the current landscape, we emphasize there are no known techniques to generate the most common mis-typings of a provided string.

5 CONCLUSIONS

The web identifiers – such as domains, hashtags, and usernames – that label our online navigation and communication all tend to share similar naming properties. Particularly in mature namespaces, the name selection process may be complicated by both technical and availability constraints. Backed by the empirical data from this work, we believe that the speed and correctness with which end users can type a name is an important property for consideration. From crowdsourced measurements we collected 51k typing times, of 28k identifiers, via a distributed workforce of 1.4k participants. Domain names were our medium for exploring identifier typeability and keystroke dynamics.

We found that a surprising amount of information can be derived from these short character sequences. Regarding name selection, we learned the shortest names are not necessarily the most typeable ones. Typing times increase linearly as a function of name growth, but it is most important that a name be composed of easily parseable keywords. Familiar character sequences tend to be most quickly typed, a property that also holds for the TLD extensions of domain names. Though our models were only moderately effective at predicting precise typing times, these simple properties proved most indicative. A continued exploration of naming keystroke dynamics led to the finding that high intra-character latencies are often indicative of word boundaries; an important tokenization clue given that many naming systems prohibit traditional delimiters.

Future work remains to understand if mobile typing behaviors mirror their desktop counterparts, and how functionalities such as auto-complete affect identifier usage and entry. However, we believe our measurements herein have applications ranging from simple advice for name-seeking individuals, to the algorithmic foundations for tools implementable by those who distribute and suggest identifiers.

REFERENCES

[1] Pieter Agten, Wouter Joosen, Frank Piessens, and Nick Nikiforakis. 2015. Seven Months' Worth of Mistakes: A Longitudinal Study of Typosquatting Abuse. In *NDSS '15: Proceedings of the ISOC Network and Distributed System Security Symposium*.

[2] Jerry R. Van Aken. 2011. A Statistical Learning Algorithm for Word Segmentation. *arXiv/CoRR* abs/1105.6162 (2011).

[3] Amazon Mechanical Turk Command Line Tools. 2016. https://requester.mturk.com/developer/tools/clt. (2016).

[4] Salil P. Banerjee and Damon L. Woodard. 2012. Biometric Authentication and Identification using Keystroke Dynamics: A Survey. *Journal of Pattern Recognition Research* 7, 1 (2012).

[5] Michael Capewell. 2005. Keyboard Comparison Applet. http://www.michaelcapewell.com/projects/keyboard/compare_applet.htm. (2005).

[6] Clayton Epp, Michael Lippold, and Regan L. Mandryk. 2011. Identifying Emotional States using Keystroke Dynamics. In *CHI '11: Proceedings of the 29th SIGCHI Conference on Human Factors in Computing Systems*.

[7] Leah Findlater, Jacob Wobbrock, and Daniel Wigdor. 2011. Typing on Flat Glass: Examining Ten-finger Expert Typing Patterns on Touch Surfaces. In *SIGCHI '11: Conf. on Human Factors in Comp. Systems*.

[8] Eibe Frank, Mark A. Hall, and Ian H. Witten. 2016. *Data Mining: Practical Machine Learning Tools and Techniques* (4th ed.). Morgan Kaufmann, Chapter The WEKA Workbench.

[9] Donald R. Gentner. 1982. Evidence Against a Central Control Model of Timing in Typing. *Journal of Experimental Psychology: Human Perception and Performance* 8, 6 (1982).

[10] Internet Corporation for Assigned Names and Numbers. 2016. Global Registrant Survey. https://newgtlds.icann.org/en/reviews/cct/global-registrant-survey-15sep16-en.pdf. (2016).

[11] Panagiotis G. Ipeirotis. 2010. Analyzing the Amazon Mechanical Turk Marketplace. *XRDS* 17, 2 (December 2010).

[12] Kevin S. Killourhy and Roy A. Maxion. 2009. Comparing Anamoly-detection Algorithms for Keystroke Dynamics. In *DSN '09: Proc. of the 39th IEEE/IFIP Intl. Conference on Dependable Systems & Networks*.

[13] Jeremy Kun. 2012. Word Segmentation, or Makingsenseofthis. https://jeremykun.com/2012/01/15/word-segmentation/. (2012).

[14] Chris Matyszczyk. 2013. Confusing Twitter Hashtag Leaves Cher Fans in Mourning. https://www.cnet.com/news/confusing-twitter-hashtag-leaves-cher-fans-in-mourning/. (2013).

[15] Fabian Monrose and Aviel Rubin. 1997. Authentication via Keystroke Dynamics. In *CCS '97: Proc. of the Conf. on Comp. and Comm. Security*.

[16] Tyler Moore and Benjamin Edelman. 2010. Measuring the Perpetrators and Funders of Typosquatting. In *FC '10: Proc. of the 14th International Conference on Financial Cryptography and Data Security*.

[17] Peter Norvig. 2009. *Beautiful Data: The Stories Behind Elegant Data Solutions*. O'Reilly Media, Chapter Natural Language Corpus Data.

[18] Uriel Priva. 2010. Constructing Typing-Time Corpora: A New Way to Answer Old Questions. In *CogSci '10: Proceedings of the 32nd Annual Meetings of the Cognitive Science Society*.

[19] Lisa P. Ramsey. 2010. Brandjacking on Social Networks: Trademark Infringement by Impersonation of Markholders. *Buffalo Law R.* (2010).

[20] Jack Reuter, Jhonata Pereira-Martins, and Jugal Kalita. 2016. Segmenting Twitter Hashtags. *Intl. J. on Natural Lang. Computing* 5, 4 (2016).

[21] Marko Robnik-Sikonja and Igor Kononenko. 1997. An Adaptation of Relief for Attribute Estimation in Regression. In *ICML '97: Proc. of the 14th International Conference on Machine Learning*.

[22] Andrea Rowland. 2015. 10 Tips for Choosing the Perfect Domain Name. https://www.godaddy.com/garage/smallbusiness/launch/10-tips-for-choosing-the-perfect-domain-name/. (2015).

[23] Shirish Shevade, S Sathiya Keerthi, Chiranjib Bhattacharyya, and Karaturi Radha Krishna Murthy. 2000. Improvements to the SMO algorithm for SVM regression. *IEEE Transac. on Neural Networks* 11, 5 (2000).

[24] SimpleCaptcha. 2011. http://simplecaptcha.sourceforge.net/. (2011).

[25] Sriram Srinivasan, Sourangshu Bhattacharya, and Rudrasis Chakraborty. 2012. Segmenting Web-domains and Hashtags Using Length Specific Models. In *CIKM '12: Proc. of the 21st ACM International Conference on Information and Knowledge Management*.

[26] New gTLD Statistics. 2017. https://ntldstats.com/. (2017).

[27] Janos Szurdi, Balazs Kocso, Gabor Cseh, Johnathan Spring, Mark Felegyhazi, and Chis Kanich. 2014. The Long Taile of Typosquatting Domain Names. In *SEC '14: Proc. of the 23rd UNIX Security Sym.*

[28] Matthew Thomas and Jasenko Ibrahimbegovic. 2015. Evaluating Typeability of Domain Names. (2015). US Patent 9,195,316.

[29] Verisign, Inc. 2016. The Domain Name Industry Brief. https://www.verisign.com/assets/domain-name-report-july2016.pdf. (2016).

[30] Kuansan Wang, Christopher Thrasher, and Bo-June Paul Hsu. 2011. Web Scale NLP: A Case Study on URL Word Breaking. In *WWW '11: Proc. of the 20th International Conference on World Wide Web*.

[31] Pascal Zesiger, Jean-Pierre Orliaguetand Louis-Jean Boe, and Pierre Mounou. 1994. *Advances in Handwriting and Drawing: A Multidisciplinary Approach*. Europia, Chapter The Influence of Syllabic Structure in Handwriting and Typing Production.