Who is .com? Learning to Parse WHOIS Records

Suqi Liu
suqi@cs.ucsd.edu

Ian Foster
idfoster@cs.ucsd.edu

Stefan Savage
savage@cs.ucsd.edu

Geoffrey M. Voelker
voelker@cs.ucsd.edu

Lawrence K. Saul
saul@cs.ucsd.edu

Department of Computer Science and Engineering
University of California, San Diego

ABSTRACT

WHOIS is a long-established protocol for querying information about the 280M+ registered domain names on the Internet. Unfortunately, while such records are accessible in a "human-readable" format, they do not follow any consistent schema and thus are challenging to analyze at scale. Existing approaches, which rely on manual crafting of parsing rules and per-registrar templates, are inherently limited in coverage and fragile to ongoing changes in data representations. In this paper, we develop a statistical model for parsing WHOIS records that learns from labeled examples. Our model is a conditional random field (CRF) with a small number of hidden states, a large number of domain-specific features, and parameters that are estimated by efficient dynamic-programming procedures for probabilistic inference. We show that this approach can achieve extremely high accuracy (well over 99%) using modest amounts of labeled training data, that it is robust to minor changes in schema, and that it can adapt to new schema variants by incorporating just a handful of additional examples. Finally, using our parser, we conduct an exhaustive survey of the registration patterns found in 102M `com` domains.

Categories and Subject Descriptors

I.5.4 [**Pattern Recognition**]: Applications—*Text processing*; C.2.3 [**Computer-Communication Networks**]: Network Operations—*Public networks*; K.4.1 [**Computer and Society**]: Public Policy Issues

General Terms

Measurement

Keywords

WHOIS; Named Entity Recognition; Machine Learning; Information Extraction

IMC'15, October 28–30, 2015, Tokyo, Japan.

DOI: http://dx.doi.org/10.1145/2815675.2815693.

1. INTRODUCTION

Most common Internet protocols today offer standardized syntax and schemas. Indeed, it is the ability to easily parse and normalize protocol fields that directly enables a broad array of network measurement research (e.g., comparing and correlating from disparate data sources including BGP route tables, TCP flow data and DNS measurements). By contrast, the WHOIS protocol—the sole source of information mapping domain names to their rich ownership and administrative context—is standard only in its transport mechanism, while the format and contents of the registration data returned varies tremendously among providers. This situation significantly hampers large-scale analyses using WHOIS data, and even those researchers who do use it commonly document the complexities and limitations in doing so [1, 7, 8, 17, 24, 25, 26]. While there are a number of open source and commercial WHOIS parsers available, the lack of an underlying data schema requires them to constantly update hand-written parsing rules or templates, both limiting coverage and increasing fragility to format changes. For example, in one recent study of domain name registration, Janos reports that the tool used (*PHPWhois*) was only able to parse registrant information for 65% of their data [25]. Overall, the situation is well summarized in a recent statement by the Coalition Against Unsolicited Commercial Email (CAUCE):

> Currently whois is offered on something of a "hobbyist" basis, particularly in TLDs that use the "thin" whois model. At one provider it will use one format, while at other times and at other providers, it will use another. This lack of consistent formatting, along with restrictive access policies, makes whois access something that's only suitable for small scale interactive "craft" access rather than being a production-ready and robust service that's appropriate for the volume of domains and other resources involved in today's domain name ecosystem [2].

In this paper, we offer a statistical, data-driven approach to tackling the WHOIS parsing problem. Since such data is designed to be "human-readable" [4], we hypothesize that modest amounts of labeled data could be sufficient to train a more general model. To this end, we show that conditional random fields (CRFs) [15]—a popular class of models for problems in statistical language processing—are particularly well-suited to the problem of parsing WHOIS records. Using a highly customized CRF, we show that 100 random training examples are sufficient to obtain over 98% parsing accuracy

on com WHOIS records and 1000 such examples brings accuracy to well over 99%. Historically, com is operated using a "thin" registry model that places no limits on format diversity and is, by far, the hardest top-level domain (TLD) to parse—yet, com accounts for 45% of all registered domains in the Internet. Moreover, we demonstrate that this model generalizes well to other TLDs and is robust in the sense that deviations or evolutions in data format are incorporated into the model with a handful of additional labeled examples. Finally, using our trained statistical parser, we systematically crawl and parse the com WHOIS registration data. Using this data set, our final contribution is to provide an initial characterization of com registration data.

2. BACKGROUND

The WHOIS protocol was introduced in the early 1980s to provide directory information for domains, people and resources (e.g., the IP address space). Over time it has become the principal mechanism by which outside stakeholders associate registered domain names with corresponding metadata such as the identity of the registrar, the registrant and contact information for administrative and technical issues. Among these uses, ICANN identifies the following: *allowing network administrators to find and fix system problems, determining domain name availability, combating inappropriate uses such as spam or fraud, facilitating the identification of trademark infringement and enhancing domain registrant accountability* [14]. Unfortunately, for a variety of reasons, WHOIS data is stored in a wide assortment of unstructured text formats and thus, while it is easily human readable, bulk systematic parsing of WHOIS data is challenging in practice. Indeed, when the National Opinion Research Center (NORC) was contracted to study WHOIS data accuracy for ICANN, they only examined 2,400 domains in com and net and admitted that "many domains need[ed] to be parsed by hand" [13].

In the remainder of this section, we briefly outline the evolution of WHOIS operations, explain how this evolution has produced the data parsing challenges faced today, and review the limitations of current approaches for addressing this problem.

2.1 History

First standardized in 1982, WHOIS was closely derived from the contemporary FINGER protocol, and defined a simple text-based request response protocol (via TCP port 43) with no formal requirements on data content or format [12].[1] At the time, all such requests were handled by a single server (src-nic.arpa), operating on behalf of a single U.S. government organization (the Defense Communications Agency) and thus there was little need for a formal data schema standard. However, with the commercial and international federation of the Internet, this model came under significant pressure. By the late 1990s, commercial domain registration had become a big business and a source of considerable conflict.[2] Ultimately, it was decided to open domain registration to competition and move overall governance to a non-governmental body, the Internet Corporation for Assigned Names and Numbers (ICANN).

[1]The second iteration of the WHOIS specification, RFC954, requests that each individual on the ARPANET or MILNET register with their full name, U.S. mailing address, ZIP code, telephone number and e-mail address. These requirements were removed in the subsequent version, RFC 3912, reflecting the Internet's increasingly international nature.

[2]This conflict reaches its apex with an anti-trust lawsuit filed against Network Solutions, who then operated all commercial registration under contract to the U.S. Department of Commerce.

2.2 Through Thick and Thin

At the core of this transition, domain registration was split into two independent functions: registries, who managed the zone files for their own top-level domains (TLDs), and registrars, who contracted with registries for the right to sell domains in their TLDs to consumers. Thus, today Verisign operates the registry for com, while GoDaddy is a registrar who sells com domains (among others) to consumers. One of the complexities of this split was how to handle WHOIS data and two distinct proposals were introduced: "thick" registries and "thin" registries. Thick registries would centrally manage all registration data and thus a WHOIS query could return all available information. By contrast, thin registries would only maintain a subset of this information (particularly the identity of the registrar, dates and status of registration, and the address of the responsible name servers). All other information, notably regarding the identity and nature of the registrant, would be maintained by the individual registrar who had been contracted for that domain. Thus, in thin registries, to obtain the full registration record is a two step process: first querying the registry for the thin record, extracting the designated registrar, and then sending an additional WHOIS query to that registrar. At the time of this change, Network Solutions opted to create a thin registry for the TLDs it managed: com, net and org.[3]

This operational distinction, thick vs. thin, indirectly impacted the diversity of schemes used for WHOIS data. Thick registries typically created a single schema (driven by the operational needs of domain provisioning), and thus each such TLD exported a single schema for all of its domains (and many such TLDs share schemas due to the use of common software). By contrast, since thin registries did not store or manage this data, their registrars could format it as they saw fit. With many more registrars than registries (there are over 1400 accredited registrars for com alone[4] and an unknown number of unaccredited resellers), this delineation of responsibilities implicitly encouraged diversity of WHOIS schemas.

With the benefit of hindsight, there is widespread sentiment that the "thick" registry model is preferable—particularly due to the key need to normalize the representation of and oversight over WHOIS data. Today all new generic TLDs (gTLDs) are required to implement the thick model, 99% of existing gTLDs registries do also, and many of the 250+ Country Code TLDs (ccTLDs) do as well.[5] Unfortunately, the few gTLDs whose registries still implement the thin model include com and net, which together comprise 45% of all registered domains (and a large majority of domains in DNS and Web searches). While there have been attempts to pressure Verisign (the registry operator for com and net) to change, and well-received proposals to completely scrap the WHOIS system altogether for a protocol with a well-defined structured data schema [20], neither have happened yet. Indeed, since Verisign's contract to manage com and net will not come up for renewal again until 2018, it seems likely that users of WHOIS data will need to accommodate this status quo for some time yet. Thus, today there are two sources of format diversity in WHOIS data: between-registry diversity for thick registries and between-registrar diversity for thin registries. In this paper, we touch on both but focus primarily on com which contains the greatest range of schema diversity due to its size and age.

[3]In 2003, the Internet Society's Public Interest Registry won the contract to manage org and switched it to a thick registry model.

[4]https://www.icann.org/registrar-reports/accredited-list.html

[5]There are few studies on ccTLD operations, but one recent study of 22 ccTLDs found that only four use the thin model [22] suggesting that this community has migrated to the thick model as well.

2.3 Parsing WHOIS

The existing approaches to WHOIS parsing are template-based and rule-based.

Template-based

Most existing parsers, including the popular `deft-whois` written in Perl, `Ruby whois` and `WhoisParser` for PHP, are template-based. These parsers first classify each domain based on their TLD and provide a per-registry parser (in many cases such parsers can be shared among registries that use a common schema). If a record calls into a thin registry such as `com`, template-based parsers will extract the designated registrar's WHOIS server address from the thin record and then parse the associated thick record using a per-registrar template. This approach is very straightforward and highly effective when a good template is available. However, they do not generalize and if a template for a particular registrar or registry is not available then they will fail completely. Moreover, they are highly fragile to variation even within the templates they parse; changing a single word in the schema or reordering field elements can easily lead to parsing failure.[6] Thus, this approach is sensitive to the number and currency of the templates available—an ongoing manual task.

To make these issues concrete, we consider one of the best such template-based parsers: Clayton's deft-whois (used in his 2014 study of WHOIS abuse [3]). The version we have, $alpha - 0.0.3$, has 6 generic templates and 575 specific templates, 403 of which were written to manage registrar-based diversity in `com`. Using $97,917$ randomly selected `com` WHOIS records, we find that 94% of our test data comes from registrars or registries that are represented by these templates.[7] However, minor changes in formats since the templates were written cause the parser to fail on the vast majority of these examples, thus reinforcing our observation that this approach is fragile in the face of change.

Rule-based

The other parsing approach, exemplified by `pythonwhois`, is to craft a more general series of rules in the form of regular expressions that are designed to match a variety of common WHOIS structures (e.g., name:value formats). If carefully constructed, such rules can still achieve high coverage with less fragility to minor changes (rules will still need to be updated for more significant changes in format structure). However, unlike template-based parsers, rule-based systems do not have a crisp failure signal (i.e., the lack of a template) and thus are more challenging to evaluate. Thus, we filtered our test data to only include those entries with a *registrant* field ($93,711$ records). When running `pythonwhois` against this corpus it correctly identifies the registrant only 59% of the time.

Summary

In general, both template and rule based parsing suffer from incompleteness and fragility. Moreover, keeping them up to date requires an ongoing investment in skilled labor. There are a number of companies that provide such services (e.g., domaintools) but even they fail to parse some domains (e.g., domaintools does not report a registrant for `albygg.com`, likely due to its unusual format).

To address these challenges, this paper introduces a statistical, data-driven approach to WHOIS parsing which at once provides greater accuracy than existing methods, less fragility to variation, and lower overhead to update (typically just one labeled example of each new format).

[6]Indeed, we see such changes in practice—with one large registrar modifying their schema significantly during the four months of WHOIS measurements we took for this paper.

[7]Using the same metric, the more popular "Ruby whois" has templates only for 63% of the test data.

3. STATISTICAL PARSING

In this section we describe our statistical model for parsing thick WHOIS records. As input to the model, we divide (or chunk) each WHOIS record into its individual lines of text. Given input of this form, the goal of parsing is to label each line of text by the type of information it provides about the registered domain (e.g., name of registrant, country of origin). The statistical model we use is known as a conditional random field (CRF), and we estimate its parameters from labeled examples of parsed records; these are records in which every line of text has been tagged (manually or otherwise) by its correct label. Once a CRF is trained from examples of this form, it can be used to parse WHOIS records that have not been previously labeled. Section 3.1 reviews the basics of CRFs, and Sections 3.2–3.3 describe specifically how we apply them to the problem of parsing WHOIS records.

3.1 Conditional random fields

A conditional random field (CRF) is a probabilistic model for mapping sequences of discrete inputs (or *tokens*) into sequences of discrete outputs (or *labels*) that have the same length. We denote a token sequence by $\mathbf{x} = (x_1, x_2, \ldots, x_T)$, where each token x_t is drawn from some (possibly infinite) alphabet, and we denote a label sequence by $\mathbf{y} = (y_1, \ldots, y_T)$, where each label y_t is drawn from some finite state space. CRFs define a powerful but tractable family of probabilistic models for the posterior distribution $\Pr(\mathbf{y}|\mathbf{x})$.

There have been many successful applications of CRFs to problems in text and natural language processing [4, 16, 23]. In this paper, we use CRFs to parse WHOIS records. Thus for our application the token x_t denotes the text on the tth (non-empty) line of the WHOIS record, and the label y_t represents the type of information on this line. We assume that line breaks are used to separate different fields of information in the WHOIS record, so that each line x_t has a unique correct label y_t. Also we do not attach labels to lines that are empty or that do not contain any alphanumeric characters.

CRFs are based on a special assumption of conditional independence. In particular, they exhibit the Markov property

$$\Pr(y_t|y_{t-1}, y_{t+1}, x_t) = \Pr(y_t|y_1, \ldots, y_T, x_1, \ldots, x_T);$$

in other words, they assume that the label y_t is conditionally independent of other inputs and labels given the input x_t and the adjacent labels $y_{t\pm1}$. This is a common assumption for text processing of human-readable documents, and it is an especially natural one for the parsing of WHOIS records. Recall that in our application, the label y_t indicates the type of information provided by the line of the WHOIS record containing the text x_t. Essentially, we are assuming that this label is strongly predicted by the text x_t and the labels of adjacent lines, and that distant lines of the WHOIS record do not provide additional information for this prediction. While CRFs are able to model the strong dependence of labels on local context, the assumption of conditional independence gives rise to extremely efficient algorithms (based on dynamic programming) for inference and parameter estimation.

The Markov property in CRFs dictates the form of the posterior distribution $\Pr(\mathbf{y}|\mathbf{x})$. Of special importance is how CRFs model the strong local dependence between labels and text. At a high level, CRFs use a large number of binary-valued "features" to indicate when certain labels and text co-occur. We use

$$f_k(y_{t-1}, y_t, x_t) \in \{0, 1\} \tag{1}$$

to denote the kth such feature, which takes as its arguments one line of text x_t, its corresponding label y_t, and in some (though not necessarily all) cases, the preceding label y_{t-1}. Each of the CRF's features is designed to test a particular property of its arguments—for example, whether the text x_t is preceded by a new line and the labels y_{t-1} and y_t are not equal (which might be likely, say, if empty lines in the WHOIS record are used to separate different blocks of information). Section 3.3 describes the features of our CRF in much greater detail. In terms of these features, the posterior distribution of the CRF is given by:

$$\Pr_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_{\boldsymbol{\theta}}(\mathbf{x})} \exp\left[\sum_t \sum_k \theta_k f_k(y_{t-1}, y_t, x_t)\right], \quad (2)$$

where the parameters θ_k (one for each feature) model the dependence[8] between labels and text, and where the denominator

$$Z_{\boldsymbol{\theta}}(\mathbf{x}) = \sum_{\mathbf{y}} \exp\left[\sum_t \sum_k \theta_k f_k(y_{t-1}, y_t, x_t)\right] \quad (3)$$

normalizes the distribution so that $\sum_{\mathbf{y}} \Pr(\mathbf{y}|\mathbf{x}) = 1$. To compute the normalization factor in eq. (3), we must sum over all possible sequences of labels, the number of which is exponential in the sequence length. However, we can perform this sum efficiently by dynamic programming; the details of this calculation can be found in the appendix.

The parameters $\boldsymbol{\theta}$ of the model can be estimated from a labeled set of parsed WHOIS records. This *training data* takes the form of R labeled WHOIS records $\{(\mathbf{x}^r, \mathbf{y}^r)\}_{r=1}^R$, consisting of one token sequence and one label sequence for each record. To estimate the parameters $\boldsymbol{\theta}$, we maximize the log-likelihood of the training data,

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{r=1}^{R} \ln \Pr_{\boldsymbol{\theta}}(\mathbf{y}^r|\mathbf{x}^r), \quad (4)$$

which measures how well the CRF predicts the correct label sequence for each WHOIS record. It can be shown that this log-likelihood is a convex function of the parameters $\boldsymbol{\theta}$. Thus we can find the optimal parameters using iterative, gradient-based methods such as L-BFGS [21].

After estimating the parameters $\boldsymbol{\theta}$, we can use the CRF to parse new WHOIS records that have not been previously labeled. Let $\mathbf{x}$ denote the non-empty lines of text in a new WHOIS record. Then using the CRF, we predict the label sequence with highest posterior probability:

$$\mathbf{y}^* = \arg\max_{\mathbf{y}} \Pr_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}). \quad (5)$$

This computation is an instance of Viterbi decoding; again details are given in the appendix.

3.2 States

The labels in our CRFs belong to a discrete state space: each of them identifies a particular type of information provided by the WHOIS record. It is typical for the fields of information in WHOIS records to appear in blocks, and for the lines within these blocks to contain details that can be viewed as more specialized subfields. Accordingly, we pursue a two-level strategy for parsing WHOIS records. First we train a CRF to parse the records into coarse, high-level blocks of information. Then for blocks that are of special interest, we train another CRF to parse the sub-fields of information within these blocks.

Our first-level CRF is designed to parse WHOIS records into the following six blocks of information:

registrar	information about the registrar, such as its name, URL, and ID.
domain	information such as domain name, name server, and domain status.
date	dates when the domain was created, when it expired, when it was last updated, etc.
registrant	name, address, phone, email, and other information about the registrant
other	administrative, billing, and technical contacts, which may or may not be identical to the registrant
null	boilerplate text and legalese, often describing claim, use, and notice

Accordingly, the discrete labels *registrar*, *domain*, *date*, *registrant*, *other*, and *null* form the state space of our first-level CRF, and the model associates each non-empty line of text x_t in the WHOIS record to a label y_t from this list. We note that thin WHOIS records provide some of this information (e.g., *registrar*, *domain*, *date*), at least for the top-level domains where such records exist. On the other hand, only thick WHOIS records provide *registrant* information, as well as listing additional contacts (*other*) that may serve as a reasonable proxy when the registrant information is missing or incomplete. We use the *null* state to label large blocks of boilerplate and otherwise uninformative text in WHOIS records.

The *registrant* information in thick WHOIS records is of special interest precisely because it is not available anywhere else. Thus we use our second-level CRF to further analyze the blocks of *registrant* information identified by the first-level CRF. Specifically, for each registrant we attempt to extract the following (self-explanatory) subfields of text:

name	*id*	*org*
street	*city*	*state*
postcode	*country*	*phone*
fax	*email*	*other*

These twelve labels form the state space of our second-level CRF: in particular, each token x_t of *registrant* information in thick WHOIS records is mapped to a more specialized label y_t from this list.

3.3 Features

The effectiveness of CRFs depends on the design of features $f_k(y_{t-1}, y_t, x_t)$ that capture the distinctive properties of different states. For the parsing of thick WHOIS records, we design these features to account for the appearance of certain words, empty spaces, and punctuation markers in each line of text. These features are based on recurring patterns of text that we describe here.

Many lines of text in thick WHOIS records contain well-defined *separators*, such as colons, tabs, or ellipses. Typically the separator is used to distinguish the *titles* of fields in the WHOIS record (e.g., `Registrant Name`) from the *values* of these fields (e.g., `John Smith`). This is also a useful distinction for our CRFs to preserve when they are parsing WHOIS records. In each line of text, we

[8] To simplify the expressions in eqs. (2–3), we have adopted a slight abuse of notation: it should be understood here, and in what follows, that the sums over k at time $t = 1$ range only over those features f_k that do not depend on the argument y_{t-1} (which does not exist at the first time step).

therefore append all words to the left of the first-appearing separator with the characters `@T` (for title); likewise, we append all words to the right of the separator with the characters `@V` (for value). If a line does not contain a separator, then we append all of its words with `@V`.

We do not attach labels to empty lines of text in WHOIS records. But we know that these empty lines, when they occur, often signal that the next block of text describes a new field of the WHOIS record. To preserve this signal, we mark whenever a non-empty line of text is preceded by one or more line breaks; we do this by prepending `NL` (for new line) to the line's observed token x_t. We use similar methods to mark other revealing forms of leading white spaces or tabs (e.g., shifts) and instances of non-alphanumeric characters (e.g., punctuation, special symbols).

With the above in mind, we can now describe in general terms the features that appear in our CRFs. To generate these features, we first compile a list of all the words (ignoring capitalization) that appear in the training set of WHOIS records. We trim words that appear very infrequently from this list, but otherwise our dictionary is quite extensive, with tens of thousands of entries. Most of the features in our CRFs simply test for the appearance of a word in a line of a text x_t with a particular label y_t. For example, here is a particular feature from our first-level CRF:

$$f_1(y_t, x_t) = \begin{cases} 1 & \text{if } x_t \text{ contains the word } \texttt{organization@T} \text{ and } y_t = registrant \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

We obtain several hundred thousand features of this general form simply by composing all the words in our dictionary with the different states of the CRF. We also generate features that test for the appearance of more general classes of words. For example, here is a particular feature from our second-level CRF:

$$f_2(y_t, x_t) = \begin{cases} 1 & \text{if } x_t \text{ contains a five-digit number and } y_t = zipcode \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

Note that the features in eqs. (6–7) only examine the label y_t for the current line of text x_t, but not the label y_{t-1} for the preceding one. However we also construct features that examine both labels. For example, here is another feature from our first-level CRF:

$$f_3(y_{t-1}, y_t, x_t) = \begin{cases} 1 & \text{if } x_t \text{ contains the word } \texttt{owner@T} \text{ and } y_{t-1} = domain \text{ and } y_t = registrant \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

It is relatively simple, in this way, to generate many binary-valued features that test for distinctive patterns of text in the different fields of WHOIS records. In total, our first-level CRF for parsing WHOIS records has nearly 1M features, and our second-level CRF for parsing more detailed *registrant* information has nearly 400K features.

The goal of learning in CRFs is to determine which of these features have predictive power—namely, to estimate, for each feature $f_k(y_{t-1}, y_t, x_t)$, the weight θ_k that appears in the posterior distribution, eq. (2). There are several well-known packages (e.g., MALLET, crfsgd, CRF++) for feature generation and learning in CRFs. For the application, we implemented our own model, with a specialized feature extraction pipeline and optimization routines such as stochastic gradient descent. We also modified a well-known implementation of the limited-memory BFGS algorithm to run in parallel for our experiments.

Label	Words
registrant	registrant@T, organization@T
registrar	registrar@T, reseller@T, www@V, SEP, NL, by@T, server@T, url@T, registered@T, whois@V, provided@T, http@V, service@T
domain	dnssec@T, status@T, domain@T, com@V, server@T, nameserver@T, unsigned@V, punycode@T, SEP, org@V, clienttransferprohibited@V, 1@V, information@T, no@V
date	date@T, updated@T, created@T, 2015@V, on@V, 2014@V, expiration@T,
other	tech@T, billing@T, administrative@T, admin@T, contact@T
null	service@T, SYM, registration@T, by@T, http@T, provided@T, for@V, of@T, for@T, the@T, more@T, cn@V, contacts@T, learn@T, here@T, is@V, whois@V, information@V, server@V

Table 1: Heavily weighted features, of the form in eq. (6), for the first-level CRF that parses WHOIS records into differently labeled blocks of information.

3.4 Model parameters

Once a CRF has been estimated from data, it can be instructive to examine the features with the highest statistical weights; roughly speaking, these are the features that the model finds most useful for prediction. We do so here for the first-level CRF described in Section 3.2. The parameters of these CRFs were learned from the labeled data set described in Section 4.

First we examine the model parameters for the simplest features, of the form in eq. (6). For each label in the CRF's state space, Table 1 lists the features with the model's highest weights. Recall that words to the left of a separator are indicated by the suffix `@T`, while other words are indicated by the suffix `@V`. On one hand, many of the results are highly intuitive: for example, the word `organization` to the left of a separator suggests that the current line of text contains information about the domain's *registrant*. On the other hand, there are associations discovered by the model that could not have been guessed beforehand. This is the power of a data-driven approach.

Next we examine the model parameters for the CRF's transition-detecting features, of the form in eq. (8). Figure 1 visualizes these features as a graph: the nodes in this graph represent the labels in the CRF's state space, and the edges are annotated by the top features that the CRF uses to detect the end of one block of information and the beginning of another. Many of these features are highly intuitive; for example, the word "created" often signifies the beginning of *date* information. (There are also features, not shown in the graph, that the CRFs uses to detect self-transitions—that is, when a block of information extends across multiple lines of the WHOIS record.)

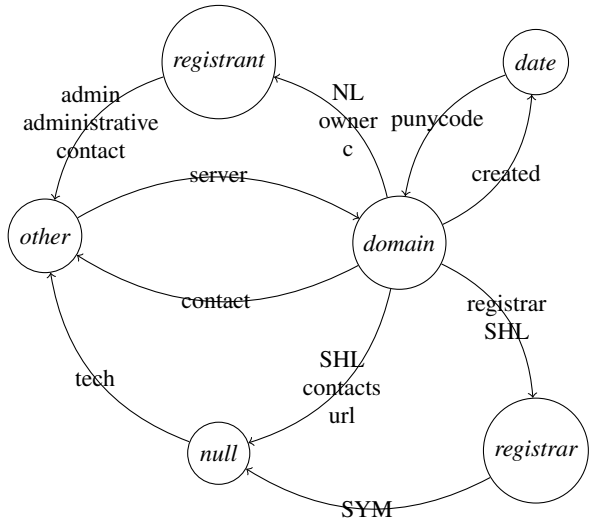


Figure 1: Visualization of predictive features for detecting adjacent blocks of information in WHOIS records. The edges of the graph show words (lower-case) and punctuation markers (upper-case) that are highly correlated with transition between differently labeled blocks. Punctuation key: NL = new line; SHL = shift left; SYM = line starts with symbols such as # or %.

4. DATA COLLECTION AND LABELING

Our dataset consists of 102M WHOIS records in the com domain (over 90% of the domains registered under the TLD), as well as some comparative samples of data from new gTLDs. We have matching thin and thick records for over 92M of these domains, but only thin records for the other cases (e.g., when a registrar server returned "no match," when our crawler was blocked, etc). In this section, we describe how the data was obtained and how we labeled a subset of it (86K randomly selected domains with thick records from com) to establish a baseline ground truth for evaluating our statistical parser.

4.1 WHOIS Crawling

Our primary dataset was obtained via long-term crawling the list of domains found in the com zone file in February of 2015. We performed an initial crawl from February to May 2015, and a second crawl from July to August 2015. As discussed earlier com is managed by Verisign under a thin registry model and thus, for each domain, there are at least two queries needed: one to Verisign to obtain the thin record and then, extracting the address of the registrar's WHOIS server from the thin record, a second query to obtain its thick record on some registrar-specific format.

The key challenge in completing this crawl is per-IP rate limiting, which we observed both at Verisign and individual registrars. Typically, once a given source IP has issued more queries to a given WHOIS server in a period than its limit, the server will stop responding, return an empty record or return an error. Queries can then resume after a penalty period is over. Unfortunately, the implementation of this rate limiting, its thresholds and triggers are rarely published publicly. This is a common problem for most research efforts that perform comprehensive online data gathering (e.g., [5, 9, 10, 18, 19]) and our solutions are not unique.

In particular, we use a simple dynamic inference technique to avoid hitting rate limits whereby we track our query rate for each WHOIS server. When a given server stops responding with valid data, we infer that our query rate was the culprit and we record this limit, subsequently querying well under this limit for that server. We use multiple servers to provide for parallel access to WHOIS servers, and we retry each query after a failure at three different servers before we mark the request as a failure.[9] We obtained 102M WHOIS records from this crawl (a bit over 90% of the com TLD). Some domains in the February 2015 zone file snapshot were expired by the time we crawled them; also, in some cases we failed for other reasons to obtain a WHOIS record (e.g., we exceeded a registrar's rate limit).

4.2 Rule-based WHOIS labeling

As we will show, our statistical parser requires only a modest amount of supervision and thus human experts are more than sufficient to source training data. However, to *evaluate* our technique requires a much larger set of ground truth data which we can compare our results to. To this end, we have manually built a rule-based parser *specifically developed* to accurately parse the thick WHOIS records of 86K com domains randomly selected from our larger corpus. As with all such systems, the resulting rule-based parser is fragile and is unlikely to generalize well outside the data it was developed for, but its purpose is to efficiently establish a large set of known results against which our own work can be benchmarked. For completeness, we describe its design here.

As with our statistical parser, our rule-based parser divides each record into line-granularity tokens. The underlying assumption, validated by our experience, is that each line encodes at most one "kind" of information. We then identify common separators (i.e., colons, spaces, tabs, etc.) that might separate any given line into "title: value" pairs (e.g., a line starting with "Registrant Name:" indicates that the text immediately following the colon is the name of the registrant). As well, we capture the common case where this relation is contextual and a field title appears alone with the following block representing the associated value (e.g., a line starting with "Registrant" might then be followed by a name, address, e-mail address and phone number). Upon this framework, we have added a large number of special case rules, iterating repeatedly until our rule-based parser was able to completely label the entries in our test corpus.

5. EVALUATION

In this section we evaluate the accuracy of the model described in Section 3. We also compare the performance of rule-based and statistical parsers that have been constructed from the same (possibly limited) set of manually labeled WHOIS records within the com domain. It is important to realize that these parsers can correctly label all the records that were used to construct them. Thus we must devise more careful ways to compare them.

We use both types of parsers to label the fields of WHOIS records as *registrar*, *domain*, *date*, *registrant*, *other*, or *null*. Then, we seek to answer three questions. First, which type of parser—rule-based or statistical—generalizes better to new WHOIS records in the same TLD? Second, how well do these parsers generalize (if at all) to WHOIS records in different TLDs? Third, when these parsers fail—presumably because they encounter WHOIS records with unfamiliar templates—how much effort is required to correct these failures? We explore each question in turn.

[9]Roughly 7.5% of domains we queries resulted in a failure after all 3 attempts.

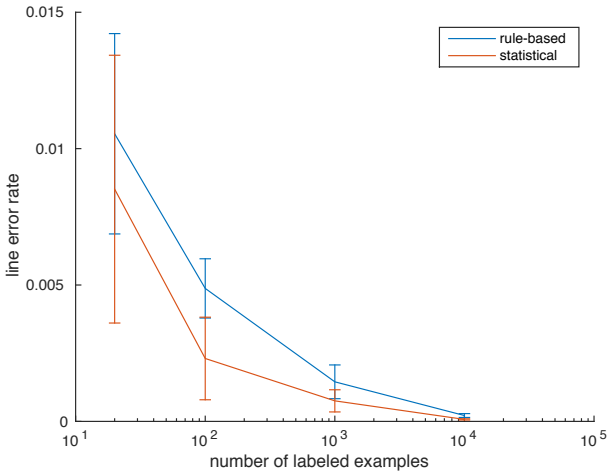


Figure 2: Line error rate versus number of labeled examples in the training set. Each point shows the average error rate from five-fold cross-validation, and each error bar shows the standard deviation across folds.

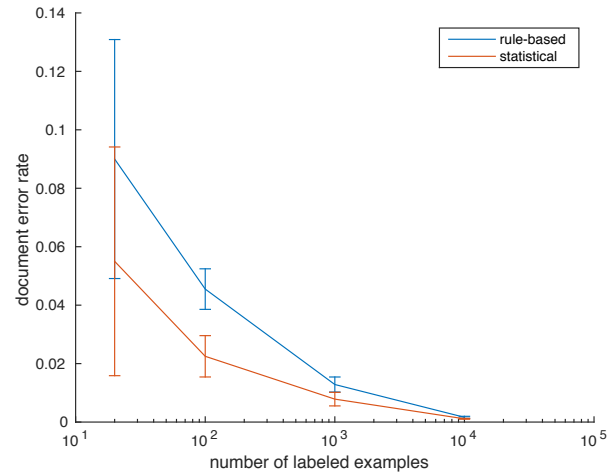


Figure 3: Document error rate versus number of labeled examples in the training set. Each point shows the average error rate from five-fold cross-validation, and each error bar shows the standard deviation across folds.

5.1 Comparison on .com

We investigate the first question, using five-fold cross-validation, on our data set of 86K labeled WHOIS records from the `com` domain. We randomly split this data set into five folds of roughly 17K records each. Within each fold we further subsample the records to obtain smaller training sets of 20, 100, 1000, and 10000 WHOIS records. Finally, we use these training sets to construct rule-based and statistical parsers, then evaluate these (purposely handicapped) parsers on the test set of the remaining WHOIS records (roughly 68K) in other folds. Thus for each training set size, we obtain five estimates of the test error, and we measure the mean and standard deviation of these estimates.

The goal of these experiments is to understand which type of parser generalizes better to new WHOIS records. To construct the statistical parsers in these experiments, we merely limit the WHOIS records that are used to estimate their model parameters. Likewise, to construct the rule-based parsers, we simply "roll back" our best rule-based parser, retaining only those rules that are necessary to label the WHOIS records in these smaller subsets. Note, however, that some pattern-matching rules cannot be rolled back, so the rule-based parser that we derive in this way is always stronger than one derived from "scratch" on the smaller subsets of WHOIS records.

Domain (Example)	*Rule-based*	*Statistical*
aero (bluemed.aero)	4/99	2/99
asia (islameyat.asia)	20/114	3/114
biz (aktivjob.biz)	36/82	0/82
coop (emheartcu.coop)	91/127	16/127
info (travelmarche.info)	0/79	0/79
mobi (amxich.mobi)	2/69	0/69
name (emrich.name)	1/28	0/28
org (fekrtna.org)	0/64	0/64
pro (olbrich.pro)	2/97	1/97
travel (tabacon.travel)	34/80	0/80
us (vc4.us)	38/88	0/88
xxx (celly.xxx)	1/66	0/66

Table 2: Comparison of parser performance in new TLDs. The columns show the fraction of mislabeled lines for a sample WHOIS record from each TLD (# error/total).

Figures 2 and 3 compare the performances of the rule-based and statistical parsers in these experiments. We measure the performance by two types of error rates on the test set: the *line* error rate, equal to the fraction of lines across all WHOIS records that are mislabeled, and the *document* error rate, equal to the fraction of records that are not perfectly labeled (i.e., in which there is at least one incorrectly labeled line). The figures show, not surprisingly, that both types of parsers improve with broader exposure to WHOIS records in `com`. However, comparing the rule-based and statistical parsers, we see that the latter dominate the former, especially when limited numbers of WHOIS records are available as labeled examples. These results suggest that the statistical parsers are learning to detect patterns of text in WHOIS records that are of broader applicability than those manually identified by the rule-based parsers. Indeed, with only 100 labeled records the statistical parser reaches an accuracy of over 97%, and with 1000 it reaches over 99%.

5.2 Comparison on new TLDs

We also compare the rule-based and statistical parsers on WHOIS records from new, unseen TLDs outside of `com`. It turns out that each of these new TLDs is owned by a single registrar, and that the WHOIS records within each TLD follow a consistent template. However, these templates are not necessarily ones that have been observed in the training set of WHOIS records from the `com` domain.

Table 2 compares the number of lines mislabeled by each type of parser on these new TLDs. In these comparisons, it is enough to sample one WHOIS record from each TLD because the formatting within each TLD is identical. There is no case in which the rule-based parser performs better than the statistical one, and there are many cases (`asia`, `biz`, `coop`, `travel`, `us`) in which it performs far worse. Again these results suggest that the statistical parser has discovered patterns of wider applicability than the rule-based one.

5.3 Comparison of maintainability

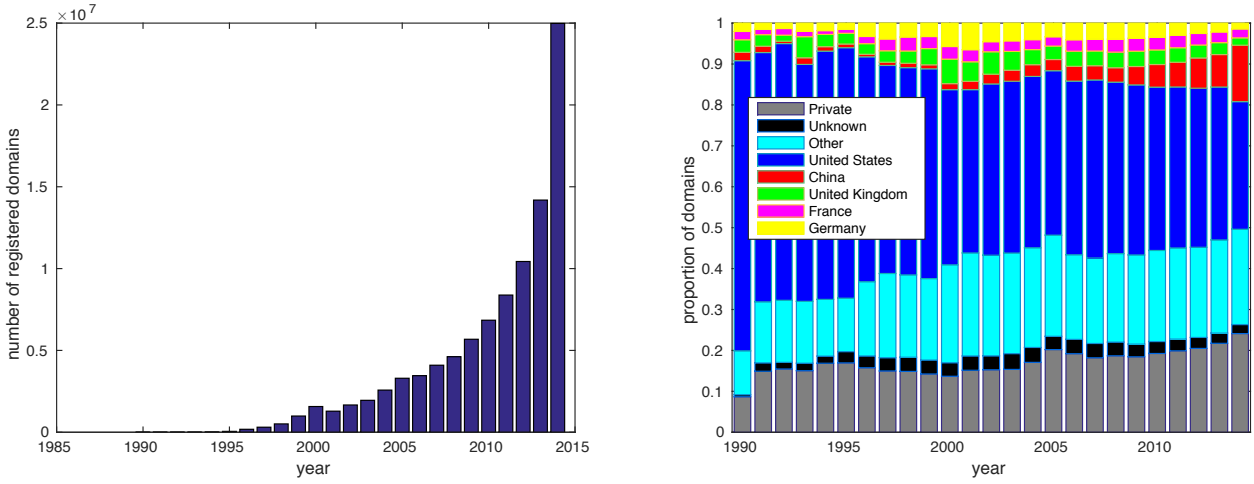
Finally we consider which type of parser is easier to maintain in an actual deployment. There are two issues here: first, how many errors are encountered when the parser is exposed to WHOIS records in different formats than it has already experienced; second, how much effort is required to fix these errors going forward?

Registrants across all time			Registrants in 2014		
Country	*Number*	*(% All)*	*(% All)*	*Number*	*Country*
United States	34,236,575	(47.6)	(41.1)	6,952,306	United States
China	6,908,865	(9.6)	(18.2)	3,072,575	China
United Kingdom	3,398,561	(4.7)	(3.5)	597,212	United Kingdom
Germany	2,518,551	(3.5)	(2.9)	482,313	France
France	2,404,450	(3.3)	(2.5)	428,306	Canada
Canada	2,152,208	(3.0)	(2.5)	426,755	India
Spain	1,480,000	(2.1)	(2.1)	356,479	Japan
Australia	1,311,191	(1.8)	(1.9)	321,504	Germany
Japan	1,242,697	(1.7)	(1.7)	293,041	Spain
India	1,143,422	(1.6)	(1.7)	293,064	Turkey
(Other)	12,609,909	(17.5)	(18.9)	3,197,172	(Other)
(Unknown)	2,458,888	(3.4)	(2.9)	482,818	(Unknown)
Total	71,865,317	(100.0)	(100.0)	16,903,545	Total

Table 3: Top 10 countries of domain registrants across all time (left) and just in 2014 (right).

Company	*Domains*
Amazon	20,596
AOL	17,136
Microsoft	16,694
21st Century Fox	14,249
Warner Bros.	13,674
Yahoo	10,502
Disney	10,342
Google	6,612
AT&T	3,931
eBay	2,570
Nike	2,566

Table 4: Well-known brand companies with the most com domains.



(a) Creation Date

(b) Country Proportions

Figure 4: Histogram of domain creation dates, and country and privacy protect breakdowns.

The results of the previous sections provide some guidance. We have already seen that the statistical parser generalizes better across familiar and unfamiliar TLDs, so that fewer errors will be encountered. Of the errors that are encountered, we can then compare the amount of effort required to fix them. For the rule-based parser, the errors can only be fixed by a human expert who is willing (on an ongoing basis) to revise the parser's existing rules or to craft altogether new ones. For the statistical parser, this manual exercise is not required: once the errors are identified, the correctly labeled WHOIS record can be added to the existing training set, and the model can be enlarged and retrained to work as desired. We emphasize here that the procedures for feature generation and parameter estimation in these models are easily automated.

We can make this comparison even more concrete for the results in Table 2. Note that the rule-based parser made errors in 10 out of the 12 new TLDs; the statistical parser made errors in just 4 of them. To fix the errors in the rule-based parser, it would be necessary for a human expert to alter the parser's rule base, one time after another, for each of these 10 TLDs. On the other hand, after retraining the model with just four additional labeled examples the resulting statistical parser has no errors.

6. SURVEYING .COM

With our parser in hand, we applied it to our crawl of the WHOIS records of com domains and constructed a database of the fields extracted by the parser. The information in the database provides a convenient global view of domain registrations in the largest TLD, and in this section we use this global perspective to look at registrations through the lenses of registrants, registrars, and the use of privacy protection. Since domain information is often used when examining Internet abuse, we also briefly look at WHOIS features of com domains found on the DBL blacklist. For the results in this section, we use 102,077,202 com domains that were created through the end of 2014.

6.1 Registrants

Where are registrants located? Table 3 shows the top 10 countries of all domain registrants, with the remaining countries combined in the "Other" row. The left half of the table shows the breakdown for all com domains, and the right half shows the breakdown for domains created in 2014. Because we have a recent snapshot of WHOIS records, we will not see domains that were registered in com

Registrations across all time			Registrations in 2014		
Registrar	*Number*	*(% All)*	*(% All)*	*Number*	*Registrar*
GoDaddy	34,932,668	(34.2)	(34.4)	8,904,002	GoDaddy
eNom	8,841,158	(8.7)	(7.7)	1,984,900	eNom
Network Solutions	5,094,458	(5.0)	(4.3)	1,111,857	Network Solutions
1&1 Internet	3,111,934	(3.0)	(3.7)	952,430	HiChina
Wild West Domains	2,636,577	(2.6)	(3.3)	846,137	Xinnet
HiChina	2,101,937	(2.1)	(3.2)	815,095	Public Domain Reg.
Public Domain Reg.	2,100,018	(2.1)	(3.0)	782,496	GMO Internet
Register.com	2,076,612	(2.0)	(2.4)	620,131	Wild West Domains
FastDomain	1,896,785	(1.9)	(2.1)	556,102	Register.com
GMO Internet	1,878,897	(1.8)	(2.1)	531,578	1&1 Internet
(Other)	37,406,158	(36.6)	(33.9)	7,714,351	(Other)
Total (All Years)	102,077,202	(100.0)	(100.0)	25,875,686	Total (2014)

Table 5: Top 10 registrars of com domains registered across all time (left) and just in 2014 (right).

Registrations using privacy protection		
Registrar	*Number*	*(% All)*
GoDaddy	6,405,390	(33.1)
eNom	2,444,342	(12.6)
GMO Internet	1,118,634	(5.8)
HiChina	764,177	(4.0)
Public Domain Reg.	644,720	(3.3)
Register.com	632,179	(3.3)
FastDomain	630,905	(3.3)
Wild West Domains	581,873	(3.0)
DreamHost	545,147	(2.8)
1&1 Internet	536,671	(2.8)
(Other)	5,021,446	(26.0)
Total	19,325,484	(100.0)

Table 6: Top 10 registrars used by privacy protected domains.

years ago and that have expired before our crawl. By also looking at just the domains registered in the last year, though, we can capture both recent registration behavior as well as a nearly complete set of domain registrations for that period (since domains typically are registered for a minimum of one year). For these results, we have also removed the 20% of all domains that use a privacy protection service since the country of the registrant cannot be inferred (we explore domains using the protection services in more detail below). For WHOIS records that do not have country information for the registrant, we list these as "Unknown".

For the WHOIS records with country information, they support the general reputation of the US dominating registrations in the com TLD: 47% of all com domains are from US registrants. Many European countries also have significant numbers of registrants, but those in China are the second most numerous. Indeed, the number of com registrants in China was nearly half those of the US in 2014, and far more than any remaining country.

Looking at these temporal trends more broadly, Figure 4a shows a histogram of the number of domains created in com over time at the granularity of a year. Figure 4b shows the same data, but normalizes it and breaks down the domains by the five largest countries of registrants as well as those using privacy protection services and registrants with missing country information.

Some general trends emerge. Registrations in com continue to grow dramatically, and the rate is increasing over time. The fraction of domains registered with privacy protection is also increasing over time, passing 20% in 2014. While US registrants dominate the total set of registered com domains, the trends are changing (at least for domains that report registrant country in their WHOIS records). The fraction of new domains from US registrants is decreasing over time, while Chinese registrants are the growth market.

Which organizations have many com domains? For the records that report organizational information, the types of organizations that stand out with the most domains are domain sellers (BuyDomains.com, HugeDomains.com, Domain Asset Holdings, etc.), online marketers (Dex Media, Yodle), and Internet hosting companies in Japan (Sakura Internet, Xserver). Beyond these, Table 4 lists well-known brand companies that have registered the most com domains. Not surprisingly, they fall into large retail, service, and media companies.[10]

[10]Note that similar company searches using services like DomainTools often return larger counts; such searches match text on the entire WHOIS record, not just particular com domains as we do.

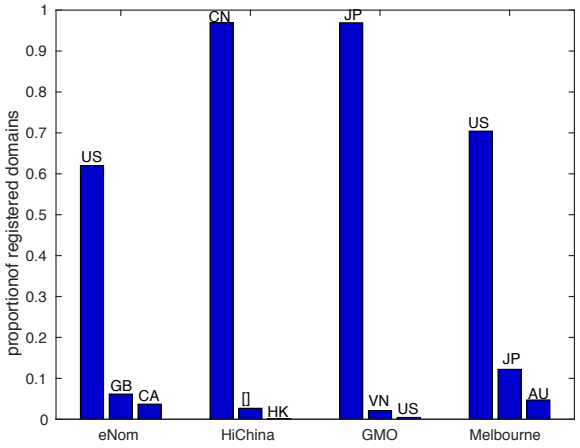


Figure 5: Top 3 registrant countries for selected registrars.

6.2 Registrars

Table 5 shows the top 10 registrars by number of com domains registered overall, and for domains created in 2014. GoDaddy is well known to be a dominant registrar, and indeed we find that it has registered over one-third of com domains. Overall registration market share is heavily skewed: the top three registrars account for nearly half of com domains, and the top 10 account for 73%. Reflecting the temporal trends in Figure 4b, we see the rise of Chinese registrars HiChina and Xin Net in the 2014 list corresponding to the rise in demand from Chinese registrants.

The registrar names evoke their countries of origin. To examine how registrants are distributed across registrars in more detail, Figure 5 shows the top three countries of registrants for four of the top registrars. For the most part, the countries of registrants reflect their registrars: eNom has US, UK, and Canadian registrants, HiChina has Chinese registrants (the "[]" corresponds to records lacking country information), and GMO Internet primarily has Japanese registrants. Interestingly, although Melbourne IT does have Australian registrants, US customers dominate its business followed by Japanese customers.

Protection Service	*Number (% All)*	
Domains By Proxy	6,901,026	(35.7)
WhoisGuard	1,336,312	(6.9)
Whois Privacy Protect	1,312,559	(6.8)
FBO REGISTRANT	945,924	(4.9)
PrivacyProtect.org	813,836	(4.2)
Aliyun	763,101	(3.9)
Perfect Privacy	651,785	(3.4)
Happy DreamHost	547,338	(2.8)
MuuMuuDomain	417,705	(2.2)
1&1 Internet	380,223	(2.0)
(Other)	5255675	(27.2)
Total	19,325,484	(100.0)

Table 7: Top 10 privacy protection services used for com domains.

Country	*Number (% All)*	
United States	32,513	(43.8)
Japan	18,630	(25.1)
China	11,882	(16.0)
Vietnam	965	(1.3)
Canada	893	(1.2)
France	860	(1.2)
India	696	(0.9)
United Kingdom	693	(0.9)
Turkey	546	(0.7)
Russia	402	(0.5)
(Other)	4,390	(5.9)
(Unknown)	1,831	(2.5)
Total	74,301	(100.0)

Table 8: Top 10 countries of registrants of com domains on the DBL in 2014.

Registrar	*Number (% All)*	
eNom	21,844	(25.1)
GoDaddy	18,085	(20.8)
GMO Internet	17,866	(20.5)
Register.com	3,880	(4.5)
Moniker	3,298	(3.8)
Network Solutions	3,164	(3.6)
Public Domain Registry	2,189	(2.5)
Xinnet	2,383	(2.7)
Name.com	2,381	(2.2)
Bizcn.com	1,991	(2.3)
(Other)	12,399	(14.2)
Total	87,099	(100.0)

Table 9: Top 10 registrars of domains on the DBL in 2014.

6.3 Privacy Protection

From Section 6.1, we saw that the use of privacy protection services is increasing over time and, overall, we found that 20% of all com domains use a privacy protection service. Table 7 shows the top 10 services as reported in the WHOIS records, both in terms of the number of com domains registered through their service and as the percentage of all domains using privacy protection. We identify privacy protection services using a small set of keywords to match against registrant name and/or organization fields in the WHOIS records, which we crafted by looking through lists of records sorted by registrants and organizations (they stand out because they by definition have many domains associated with them).

The most prominent privacy protection service is Domains By Proxy, which is owned by the founder of the GoDaddy registrar and accounts for 36% of protected com domains. Although there is a long tail of service names, the top 10 account for 73% of protected domains. However, the names used in the WHOIS records for protected domains do not always correspond to organizations that we could identify (Private Registration, Hidden by Whois Privacy Protection Service). This behavior suggests that exploring the use, identity, and operation of protection services in further detail, in addition to the registrants of those services [3], could be an interesting open topic.

From a different perspective, Table 6 shows the top 10 registrars through which privacy protection domains have been registered. The registrars used largely track the list of all com domains in Table 5.[11]

6.4 Blacklisted Domains

Domain blacklists are a common source of identifying abusive domains. As a final survey, we look at the WHOIS records for com domains that appear on the Domain Block List (DBL).[12] Note that the DBL blacklist is populated from domains that appear in spam. Because domains in other TLDs are often cheaper than com, com domains only represents a portion (46%) of all domains in the DBL. But doing the analysis provides some insight into trends revealed from looking at WHOIS features. Also, we focus on just those domains created in 2014 to minimize domain expiration; 58.8% of com domains on the DBL were created in 2014, so this set represents the bulk of com domains on the DBL.

Table 8 shows the top 10 registrant countries for com domains on the DBL. Comparing with the countries for all domains in Table 3, both the percentages and rank orderings have notable differences for domains on the blacklist: in particular, registrants from Japan, China, and Vietnam are much more pronounced.

From another perspective, Table 9 shows the top 10 registrars through which com domains on the DBL are registered. Similarly comparing with the registrars for all domains in Table 5, the results have some interesting differences. Registrars that have been implicated in abuse (eNom, Xin Net) are more prominent, and new registrars appear on the top list (Moniker, evoPlus, Bizcn.com, etc.).

These results suggest that, in addition to registrar, country information would likely be a useful feature, e.g., for predicting domains used in abuse [6, 11].

[11]Note that our crawler exceeded the rate limit for Network Solutions domains, so we only have their thin records and cannot report the prevalence of privacy protection in Network Solutions.

[12]https://www.spamhaus.org/dbl

7. CONCLUSION

In this paper we have developed a statistical parser for WHOIS records that learns a general model from labeled examples, is easy to maintain and achieves extremely high accuracy in practice. We demonstrate its utility by providing an initial survey of registration patterns in the com TLD. Finally, code and data from our study is available at: http://www.sysnet.ucsd.edu/projects/whois.

Acknowledgments

We would like to thank our shepherd Walter Willinger for his valuable guidance, and the anonymous reviewers for their helpful suggestions. We are also very grateful to Neha Chachra and Chris Grier for their assistance with the proxy network, and we are particularly indebted to Paul Pearce for going above, beyond and beyond again in our time of need. We would also like to thank Cindy Moore for managing the software and systems used for this project. This work was supported by National Science Foundation grant NSF-1237264 and by generous research, operational and/or in-kind support from Google, Microsoft, Yahoo, and the UCSD Center for Networked Systems (CNS).

8. REFERENCES

[1] X. Cai, J. Heidemann, B. Krishnamurthy, and W. Willinger. Towards an AS-to-Organization Map. In *Proceedings of the 10th ACM/USENIX Internet Measurement Conference (IMC)*, Nov. 2010.

[2] CAUCE. Submission to ICANN WHOIS Team review. `http://www.cauce.org/2011/04/submission-to-icann-whois-team-review.html`, Apr. 2011.

[3] R. Clayton and T. Mansfield. A Study of Whois Privacy and Proxy Service Abuse. In *Proceedings of the 13th Workshop on Economics of Information Security (WEIS)*, June 2014.

[4] L. Daigle. RFC 3912: WHOIS Protocol Specification. *IETF*, Sept. 2004.

[5] Z. Durumeric, E. Wustrow, and J. A. Halderman. ZMap: Fast Internet-Wide Scanning and its Security Applications. In *Proceedings of the 22nd USENIX Security Symposium*, Aug. 2013.

[6] M. Felegyhazi, C. Kreibich, and V. Paxson. On the Potential of Proactive Domain Blacklisting. In *Proceedings of the USENIX Workshop on Large-scale Exploits and Emergent Threats (LEET)*, San Jose, CA, Apr. 2010.

[7] I. Fette, N. Sadeh, and A. Tomasic. Learning to Detect Phishing Emails. In *Proceedings of the International World Wide Web Conference*, May 2007.

[8] T. Frosch. Mining DNS-related Data for Suspicious Features. Master's thesis, Ruhr Universitat Bochum, 2012.

[9] M. Gabielkov and A. Legout. The Complete Picture of the Twitter Social Graph. In *ACM CoNEXT 2012 Student Workshop*, Dec. 2012.

[10] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Y. Zhao. Detecting and Characterizing Social Spam Campaigns. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement (IMC)*, 2010.

[11] S. Hao, M. Thomas, V. Paxson, N. Feamster, C. Kreibich, C. Grier, and S. Hollenbeck. Understanding the Domain Registration Behavior of Spammers. In *Proceedings of the 13th ACM/USENIX Conference on Internet Measurement (IMC)*, 2013.

[12] K. Harrenstien, M. Stahl, and E. Feinler. RFC 812: NICNAME/WHOIS. *IETF*, Mar. 1982.

[13] ICANN. Draft Report for the Study of the Accuracy of WHOIS Registrant Contact Information. `https://www.icann.org/en/system/files/newsletters/whois-accuracy-study-17jan10-en.pdf`, Jan. 2010.

[14] ICANN. Policy Issue Brief — gTLD WHOIS. `https://www.icann.org/resources/pages/whois-2012-06-14-en`, June 2012.

[15] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML)*, 2001.

[16] A. McCallum and W. Li. Early Results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-enhanced Lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning (CONLL)*, 2003.

[17] D. K. McGratn and M. Gupta. Behind Phishing: An Examination of Phisher Modi Operandi. In *Proceedings of the USENIX Workshop on Large-scale Exploits and Emergent Threats (LEET)*, Apr. 2008.

[18] A. Mislove, H. S. Koppula, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Growth of the Flickr Social Network. In *Proceedings of the 1st ACM SIGCOMM Workshop on Social Networks (WOSN)*, Aug. 2008.

[19] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and Analysis of Online Social Networks. In *Proceedings of the 5th ACM/USENIX Internet Measurement Conference (IMC)*, Oct. 2007.

[20] A. Newton and S. Hollenbeck. Registration Data Access Protocol Query Format: Draft Standard. `https://tools.ietf.org/html/draft-ietf-weirds-rdap-query-18`, Dec. 2014.

[21] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 2nd edition, 2006.

[22] Nominet. Contact Data Disclosure in the .uk WHOIS: Appendix I. `http://www.nominet.org.uk/sites/default/files/Appendix-I-Comparative-registry-and-WHOIS-data-publication-review.pdf`, 2015.

[23] F. Sha and F. Pereira. Shallow Parsing with Conditional Random Fields. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL)*, 2003.

[24] T. Stallings, B. Wardman, G. Warner, and S. Thapaliya. "WHOIS" Selling All The Pills. *International Journal of Forensic Science*, 7(2):46–63, 2012.

[25] J. Szurdi. Understanding the Purpose of Domain Registrations. Master's thesis, Budapest University of Technology and Economics, 2012.

[26] T. Vissers, W. Joosen, and N. Nikiforakis. Parking Sensors: Analyzing and Detecting Parking Domains. In *Proceedings of the Network and Distributed System Security Symposum (NDSS)*, Feb. 2015.

APPENDIX

A. PROBABILISTIC INFERENCE

In this appendix we describe the essential computations for probabilistic inference in CRFs. By exploiting the Markov property in CRFs, we can efficiently compute both the log-likelihood in eq. (4) and the most likely label sequence in eq. (5).

First, we consider the normalization factor in eq. (3), whose computation requires summing over the exponentially many label sequences of length T. Let n denote the number of states in the CRF. As a useful shorthand, we define an $n \times n$ matrix $\mathbf{M_t}$ at each time step t, whose elements are given by:

$$[\mathbf{M_t}]_{ij} = \exp\left[\sum_k \theta_k f_k(y_{t-1}=i, y_t=j, x_t)\right]. \quad (9)$$

As previously noted, it is to be understood in eq. (9) that the sum over k at time $t=1$ only ranges over those features that do not have any dependence on the argument y_{t-1}. It is then a straightforward exercise to show that the normalization factor in eq. (3) is given by:

$$Z_\theta(\mathbf{x}) \;=\; \sum_{ij} \left[\mathbf{M_1 M_2} \ldots \mathbf{M_T}\right]_{ij}. \quad (10)$$

This computation, which involves T matrix-vector products, can be performed in $O(n^2 T)$ operations.

Once the normalization factor in eq. (10) is computed, the log-likelihood in eq. (4) follows immediately. In particular, substituting eq. (2) into eq. (4), we obtain:

$$\mathcal{L}(\theta) \;=\; \sum_{r=1}^{R}\left[\sum_{t,k} \theta_k f_k(y^r_{t-1}, y^r_t, x^r_t) - \log Z_\theta(\mathbf{x}^r)\right]. \quad (11)$$

Eq. (11) gives the log-likelihood $\mathcal{L}(\theta)$ explicitly as a function of the model parameters θ. In order to optimize these parameters, it is of course necessary to compute the gradient $\frac{\partial \mathcal{L}}{\partial \theta}$. This can be done using the forward-backard algorithm for CRFs, which involves just a simple extension of the procedure for computing the normalization factor $Z_\theta(\mathbf{x})$. In particular, the elements of the gradient are closely related to the marginal probabilities of the distribution in eq. (2). But these, for example, are given simply by:

$$\begin{aligned}&\Pr_\theta(y_{t-1}=\ell, y_t=m|\mathbf{x}) \\ &\quad = \quad \frac{1}{Z_\theta(\mathbf{x})}\sum_{ij}\left[\mathbf{M_1}\ldots\mathbf{M_{t-1}}\right]_{i\ell}\left[\mathbf{M_t}\ldots\mathbf{M_T}\right]_{mj}. \quad (12)\end{aligned}$$

We refer the reader to the classic treatment [15] of CRFs for more details.

Finally we show how to compute the most likely sequence of labels in eq. (5). The computation is simplified by working in the log-domain and noting that the normalization factor $Z_\theta(\mathbf{x})$ is independent of $\mathbf{y}$. From these considerations we obtain the simpler expression:

$$\mathbf{y}^* \;=\; \arg\max_{\mathbf{y}}\left[\sum_{t,k}\theta_k f_k(y_{t-1}, y_t, x_t)\right]. \quad (13)$$

The computation in eq. (13) is a straightforward exercise in dynamic programming. We introduce an $n \times T$ matrix $\mathbf{V}$ and define its first column by:

$$V_{i1} = \sum_k \theta_k f_k(y_1=i, x_1). \quad (14)$$

Then we fill in the matrix elements recursively, one column at a time, as follows:

$$V_{jt} = \max_i\left[V_{i,t-1} + \sum_k \theta_k f_k(i, j, x_t)\right]. \quad (15)$$

In terms of this matrix, the most likely label at time T is simply by $y^*_T = \arg\max_i V_{iT}$. To derive the most likely labels at earlier times, we only need to record the index used at each step of the recursion in eq. (15). In particular, let

$$\textsc{index}_t(j) = \arg\max_i\left[V_{i,t-1} + \sum_k \theta_k f_k(i, j, x_t)\right]. \quad (16)$$

Then in terms of these indices, the most likely labels at all earlier times are derived from the backtracking procedure:

$$y^*_t = \textsc{index}_{t+1}(y^*_{t+1}). \quad (17)$$

Finally we note that this computation for the most likely labels y^*_t can also be performed in $O(n^2 T)$ operations.