Learning to Identify Regular Expressions that Describe Email Campaigns (Online Appendix)

Paul Prasse PRASSE@CS.UNI-POTSDAM.DE
Christoph Sawade SAWADE@CS.UNI-POTSDAM.DE
Niels Landwehr LANDWEHR@CS.UNI-POTSDAM.DE
Tobias Scheffer SCHEFFER@CS.UNI-POTSDAM.DE
University of Potsdam, Department of Computer Science, August-Bebel-Strasse 89, 14482 Potsdam, Germany

A. Definitions

Definition 2 (Regular Expressions)**.** The set $\mathcal{Y}_\Sigma$ of regular expressions over an ordered alphabet Σ is recursively defined as follows.

- Every $\mathbf{y}_j \in \Sigma \cup \{\epsilon, ., \backslash\mathsf{S}, \backslash\mathsf{e}, \backslash\mathsf{w}, \backslash\mathsf{d}\}$, every range $\mathbf{y}_j = l_{min}\text{–}l_{max}$, where $l_{min}, l_{max} \in \Sigma$ and $l_{min} < l_{max}$, and their disjunction $[\mathbf{y}_1 \ldots \mathbf{y}_k]$ are regular expressions.
- If $\mathbf{y}_1, \ldots, \mathbf{y}_k \in \mathcal{Y}_\Sigma$ are regular expressions, so are the concatenation $\mathbf{y} = \mathbf{y}_1 \ldots \mathbf{y}_k$, the disjunction $\mathbf{y} = \mathbf{y}_1 | \ldots | \mathbf{y}_k$, $\mathbf{y} = \mathbf{y}_1?$, $\mathbf{y} = (\mathbf{y}_1)$, and the repetitions $\mathbf{y} = \mathbf{y}_1^*$, $\mathbf{y} = \mathbf{y}_1^+$, $\mathbf{y} = \mathbf{y}_1\{l\}$, and $\mathbf{y} = \mathbf{y}_1\{l, u\}$, where $l, u \in \mathbb{N}$ and $l \leq u$.

We now define the syntax tree, the parse tree, and the matching lists for a regular expression $\mathbf{y}$ and a string $x \in \Sigma^*$. The shorthand $(\mathbf{y} \rightarrow T_1, \ldots, T_k)$ denotes the tree $T = (V, E, \Gamma, \leq)$ with root node $v_0 \in V$ labeled with $\Gamma(v_0) = \mathbf{y}$ and subtrees $T_1, \ldots, T_k$. The order $\leq$ maintains the subtree orderings $\leq_i$ and defines the root node as the minimum over the set V and $v' \leq v''$ for all $v' \in V_i$ and $v'' \in V_j$, where $i < j$.

Definition 3 (Syntax Tree)**.** The abstract syntax tree $T^{\mathbf{y}}_{syn}$ for a regular expression $\mathbf{y}$ is recursively defined as follows. Let $T^{\mathbf{y}_j}_{syn} = (V^{\mathbf{y}_j}_{syn}, E^{\mathbf{y}_j}_{syn}, \Gamma^{\mathbf{y}_j}_{syn}, \leq^{\mathbf{y}_j}_{syn})$ be the syntax tree of the subexpression $\mathbf{y}_j$.

- If $\mathbf{y} \in \Sigma \cup \{\epsilon, ., \backslash\mathsf{S}, \backslash\mathsf{e}, \backslash\mathsf{w}, \backslash\mathsf{d}\}$, or if $\mathbf{y} = l_{min}\text{–}l_{max}$,
  where $l_{min}, l_{max} \in \Sigma$, we define
  $T^{\mathbf{y}}_{syn} = (\mathbf{y} \rightarrow \emptyset)$.
- If $\mathbf{y} = (\mathbf{y}_1)$,
  where $\mathbf{y}_1 \in \mathcal{Y}_\Sigma$, we define
  $T^{\mathbf{y}}_{syn} = T^{\mathbf{y}_1}_{syn}$.
- If $\mathbf{y} = \mathbf{y}_1^*$, $\mathbf{y} = \mathbf{y}_1^+$,
  $\mathbf{y} = \mathbf{y}_1\{l, u\}$, or if $\mathbf{y} = \mathbf{y}_1\{l\}$,
  where $\mathbf{y}_1 \in \mathcal{Y}_\Sigma$, $l, u \in \mathbb{N}$, and there exist no $\mathbf{y}', \mathbf{y}'' \in \mathcal{Y}_\Sigma$ such that $\mathbf{y}_1 = \mathbf{y}'|\mathbf{y}''$ or $\mathbf{y}_1 = \mathbf{y}'\mathbf{y}''$, we define
  $T^{\mathbf{y}}_{syn} = (\mathbf{y} \rightarrow T^{\mathbf{y}_1}_{syn})$.
- If $\mathbf{y} = \mathbf{y}_1 \ldots \mathbf{y}_k$,
  where $\mathbf{y}_j \in \mathcal{Y}_\Sigma$, and there exist no $\mathbf{y}', \mathbf{y}'' \in \mathcal{Y}_\Sigma$ such that $\mathbf{y}_j = \mathbf{y}'|\mathbf{y}''$ or $\mathbf{y}_j = \mathbf{y}'\mathbf{y}''$, we define
  $T^{\mathbf{y}}_{syn} = (\mathbf{y} \rightarrow T^{\mathbf{y}_1}_{syn}, \ldots, T^{\mathbf{y}_k}_{syn})$.
- If $\mathbf{y} = \mathbf{y}_1 | \ldots | \mathbf{y}_k$,
  where $\mathbf{y}_j \in \mathcal{Y}_\Sigma$, and there exist no $\mathbf{y}', \mathbf{y}'' \in \mathcal{Y}_\Sigma$ such that $\mathbf{y}_j = \mathbf{y}'|\mathbf{y}''$, or if
  $\mathbf{y} = [\mathbf{y}_1 \ldots \mathbf{y}_k]$ and there exist no $\mathbf{y}', \mathbf{y}'' \in \mathcal{Y}_\Sigma$ such that $\mathbf{y}_j = \mathbf{y}'\mathbf{y}''$, we define
  $T^{\mathbf{y}}_{syn} = (\mathbf{y} \rightarrow T^{\mathbf{y}_1}_{syn}, \ldots, T^{\mathbf{y}_k}_{syn})$.

Definition 4 (Parse Tree and Matching List)**.** Given a syntax tree $T^{\mathbf{y}}_{syn} = (V^{\mathbf{y}}_{syn}, E^{\mathbf{y}}_{syn}, \Gamma^{\mathbf{y}}_{syn}, \leq^{\mathbf{y}}_{syn})$ of a regular expression $\mathbf{y}$ with nodes $v \in V^{\mathbf{y}}_{syn}$ and a string $x \in L(\mathbf{y})$, a parse tree $T^{\mathbf{y},x}_{par}$ and the matching lists $M^{\mathbf{y},x}(v)$ for each $v \in V^{\mathbf{y}}_{syn}$ are recursively defined as follows. Let $T^{\mathbf{y}_j,x}_{par} = (V^{\mathbf{y}_j,x}_{par}, E^{\mathbf{y}_j,x}_{par}, \Gamma^{\mathbf{y}_j,x}_{par}, \leq^{\mathbf{y}_j,x}_{par})$ be the parse tree and $T^{\mathbf{y}_j}_{syn} = (V^{\mathbf{y}_j}_{syn}, E^{\mathbf{y}_j}_{syn}, \Gamma^{\mathbf{y}_j}_{syn}, \leq^{\mathbf{y}_j}_{syn})$ the syntax tree of the subexpression $\mathbf{y}_j$.

- If $\mathbf{y} = x$ and $x \in \Sigma \cup \{\epsilon\}$, we define
  $M^{\mathbf{y},x}(v_0) = \{x\}$ and
  $T^{\mathbf{y},x}_{par} = (\mathbf{y} \rightarrow \emptyset)$.
- If $\mathbf{y} = .$ and $x \in \Sigma$,
  $\mathbf{y} = l_{min}\text{–}l_{max}$ and $l_{min} \leq x \leq l_{max}$, or if
  $\mathbf{y} \in \{\backslash\mathsf{S}, \backslash\mathsf{w}, \backslash\mathsf{e}, \backslash\mathsf{d}\}$ and x is either a non-whitespace character (everything but spaces, tabs, and line breaks), a word character (letters, digits, and underscores), a character in $\{., -, \#, +\}$ or a word character, or a digit,

respectively, we define
$M^{\mathbf{y},x}(v) = \{x\}$ for all $v \in V^{\mathbf{y}}_{syn}$ and
$T^{\mathbf{y},x}_{par} = (\mathbf{y} \rightarrow T^{x,x}_{par})$.

- If $\mathbf{y} = (\mathbf{y}_1)$ and $x \in \Sigma^*$, we define
$M^{\mathbf{y},x}(v) = M^{\mathbf{y}_1,x}(v)$ for all $v \in V^{\mathbf{y}}_{syn}$ and
$T^{\mathbf{y},x}_{par} = T^{\mathbf{y}_1,x}_{par}$

- If $\mathbf{y} = \mathbf{y}_1^*$, $x = x_1 \dots x_k$, and $k \geq 0$, or if
$\mathbf{y} = \mathbf{y}_1^+$, and $k > 0$, or if
$\mathbf{y} = \mathbf{y}_1\{l, u\}$, and $l \leq k \leq u$, or if
$\mathbf{y} = \mathbf{y}_1\{l\}$, and $k = l$,
where $x_i \in \Sigma^+$, and there exist no $\mathbf{y}', \mathbf{y}'' \in \mathcal{Y}_\Sigma$ such that $\mathbf{y}_1 = \mathbf{y}'|\mathbf{y}''$ or $\mathbf{y}_1 = \mathbf{y}'\mathbf{y}''$, we define
$M^{\mathbf{y},x}(v) = \begin{cases} \{x\} & \text{, if } v = v_0 \\ \bigcup_{i=1}^k M^{\mathbf{y}_1,x_i}(v) & \text{, if } v \in V^{\mathbf{y}_1}_{syn} \end{cases}$, and
$T^{\mathbf{y},x}_{par} = (\mathbf{y} \rightarrow T^{\mathbf{y}_1,x_1}_{par}, \dots, T^{\mathbf{y}_1,x_k}_{par})$.

- If $\mathbf{y} = \mathbf{y}_1 \dots \mathbf{y}_k$, $x = x_1 \dots x_k$,
where $x_i \in \Sigma^*$, and there exist no $\mathbf{y}', \mathbf{y}'' \in \mathcal{Y}_\Sigma$ such that $\mathbf{y}_j = \mathbf{y}'|\mathbf{y}''$ or $\mathbf{y}_j = \mathbf{y}'\mathbf{y}''$, we define
$M^{\mathbf{y},x}(v) = \begin{cases} \{x\} & \text{, if } v = v_0 \\ M^{\mathbf{y}_j,x_i}(v) & \text{, if } v \in V^{\mathbf{y}_j}_{syn} \end{cases}$, and
$T^{\mathbf{y},x}_{par} = (\mathbf{y} \rightarrow T^{\mathbf{y}_1,x_1}_{par}, \dots, T^{\mathbf{y}_k,x_k}_{par})$.

- If $\mathbf{y} = \mathbf{y}_1 | \dots | \mathbf{y}_k$, $x \in \Sigma^*$
and there exist no $\mathbf{y}', \mathbf{y}'' \in \mathcal{Y}_\Sigma$ such that $\mathbf{y}_j = \mathbf{y}' \mid \mathbf{y}''$, or if
$\mathbf{y} = [\mathbf{y}_1 \dots \mathbf{y}_k]$, $x \in \Sigma^+$
and there exist no $\mathbf{y}', \mathbf{y}'' \in \mathcal{Y}_\Sigma$ such that $\mathbf{y}_j = \mathbf{y}'\ \mathbf{y}''$, we define
$M^{\mathbf{y},x}(v) = \begin{cases} \{x\} & \text{, if } v = v_0 \\ M^{\mathbf{y}_j,x}(v) & \text{, if } v \in V^{\mathbf{y}_j}_{syn} \text{, and} \\ \emptyset & \text{, otherwise} \end{cases}$
$T^{\mathbf{y},x}_{par} = (\mathbf{y} \rightarrow T^{\mathbf{y}_j,x}_{par})$.

If $x \notin L(\mathbf{y})$, that is, no parse tree can be derived by the specification above, the empty sets $M^{\mathbf{y},x}(v) = \emptyset$ for all $v \in V^{\mathbf{y}}_{syn}$ and $T^{\mathbf{y},x}_{par} = \emptyset$ are returned. Otherwise, we denote the set of all parse trees and the unions of all matching lists for each $v \in V^{\mathbf{y}}_{syn}$ satisfying Definition 4 by $\mathcal{T}^{\mathbf{y},x}_{par}$ and $\mathcal{M}^{\mathbf{y},x}(v)$, respectively. Finally, the matching list $M^{\mathbf{y},\mathbf{x}}(v)$ for a set of strings $\mathbf{x}$ for node $v \in V^{\mathbf{y}}_{syn}$ is defined as $M^{\mathbf{y},\mathbf{x}}(v) = \bigcup_{x \in \mathbf{x}} \mathcal{M}^{\mathbf{y},x}(v)$.

B. Used Features

Let M be a matching list and M_Σ be the set of characters in Σ which appear in the matching list M. The list of binary and continuous features, used to train *REx-SVM* is shown in Table 1.

Table 1. Important features used to train *REx-SVM*.

Feature	**Description**
$[\![\varepsilon \in M]\!]$	Matching list contains the empty string?
$[\![\forall \mathbf{x} \in M \lvert\mathbf{x}\rvert = 1]\!]$	All elements of the matching list have the length one?
$[\![\exists i \in \mathbb{N} \forall \mathbf{x} \in M \lvert\mathbf{x}\rvert = i]\!]$	All elements of the matching list have the same length?
$\frac{\lvert M_\Sigma \cap \{A,\ldots,Z\}\rvert}{26}$	Portion of characters A–Z in the matching list
$\frac{\lvert M_\Sigma \cap \{a,\ldots,z\}\rvert}{26}$	Portion of characters a–z in the matching list
$\frac{\lvert M_\Sigma \cap \{0,\ldots,9\}\rvert}{10}$	Portion of characters 0–9 in the matching list
$\frac{\lvert M_\Sigma \cap \{A,\ldots,F\}\rvert}{6}$	Portion of characters A–F in the matching list
$\frac{\lvert M_\Sigma \cap \{a,\ldots,f\}\rvert}{6}$	Portion of characters a–f in the matching list
$\frac{\lvert M_\Sigma \cap \{G,\ldots,Z\}\rvert}{20}$	Portion of characters G–Z in the matching list
$\frac{\lvert M_\Sigma \cap \{g,\ldots,z\}\rvert}{20}$	Portion of characters g–z in the matching list
$[\![\forall x \in M_\Sigma x \notin \{A,\ldots,Z\}\ ]\!]$	No characters of A–Z in the matching list?
$[\![\forall x \in M_\Sigma x \notin \{a,\ldots,z\}\ ]\!]$	No characters of a–z in the matching list?
$[\![\forall x \in M_\Sigma x \notin \{0,\ldots,9\}\ ]\!]$	No characters of 0–9 in the matching list?
$[\![\forall x \in M_\Sigma x \notin \{a,\ldots,f\}\ ]\!]$	No characters of a–f in the matching list?
$[\![\forall x \in M_\Sigma x \notin \{A,\ldots,F\}\ ]\!]$	No characters of A–F in the matching list?
$[\![\lvert M_\Sigma \cap \{-,/,?,=,.,@,:\}\rvert > 0]\!]$	Matching list contains URL/Email characters?
$[\![\forall \mathbf{x} \in M \lvert\mathbf{x}\rvert \geq 1 \wedge \lvert\mathbf{x}\rvert \leq 5]\!]$	Length of strings in the matching list is between 1 and 5?
$[\![\forall \mathbf{x} \in M \lvert\mathbf{x}\rvert \geq 6 \wedge \lvert\mathbf{x}\rvert \leq 10]\!]$	Length of strings in the matching list is between 5 and 10?
$[\![\forall \mathbf{x} \in M \lvert\mathbf{x}\rvert \geq 11 \wedge \lvert\mathbf{x}\rvert \leq 20]\!]$	Length of strings in the matching list is between 10 and 20?
$[\![\forall \mathbf{x} \in M \lvert\mathbf{x}\rvert > 20]\!]$	Length of strings in the matching list is higher than 20?
$[\![\lvert M\rvert = 0]\!]$	Matching list is empty?