Improving Real-world Password Guessing Attacks via Bi-directional Transformers

Ming Xu, Jitao Yu, Xinyi Zhang, Chuanwang Wang,
Shenghao Zhang, Haoqi Wu, and Weili Han

Fudan University
Facebook

Passwords are widely prevalent

* *******

Passwords Guessing Attacks

No extra information

General guessing attacks

Markov-based, PCFG-based

Hashcat

Target passwords

Getting extra information

Real-world guessing attacks

Targeted guessing attacks

Three Real-world Guessing Scenarios

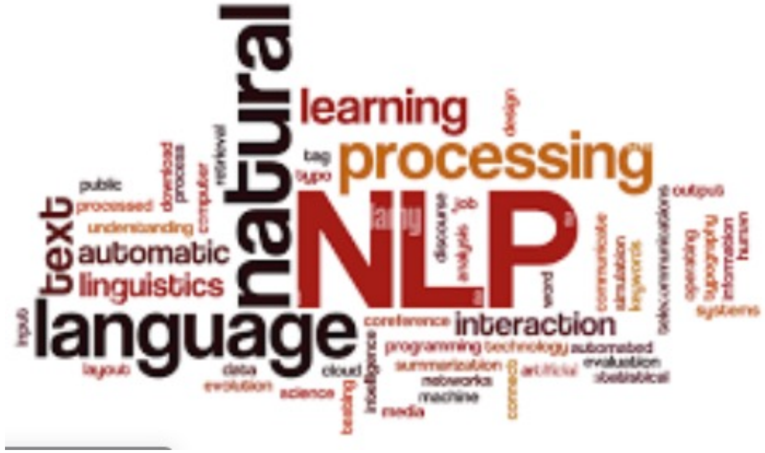
Conditional Password Guessing (CPG) → p***w0rd →p@ssw0rd → **CWAE** auto-encoders [Pasquini et al, SP-2021]

Targeted Password Guessing (TPG) → Alice1997→@lice197! → **Pass2path** RNN [Pal et al, SP-2019]

Adaptive Rule-based Password Guessing (ARPG) → (A->@) → password →p@ssword → **ADaMs** CNN [Pasquini et al, USENIX-2021]

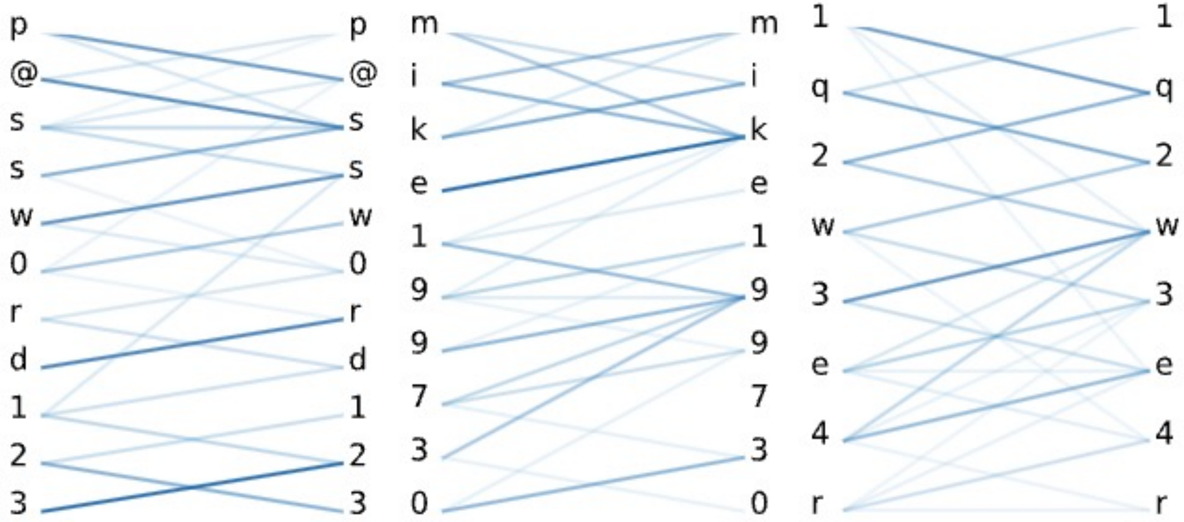
Password Guessing Attacks can benefit from techniques in natural language

* * * * * * * *



Bi-directional transformers

Pre-trained framework



Contributions

- ❑ We propose a bi-directional-transformer-based framework that uses the pre-training and fine-tuning paradigm in password guessing domain.
- ❑ With our pre-trained framework, we design three attack-specific fine-tuning approaches for CPG, TPG and ARPG.
- ❑ We introduce a hybrid password strength meter (HPSM) with sub-second latency to mitigate these risks from real-world.

Design Challenges

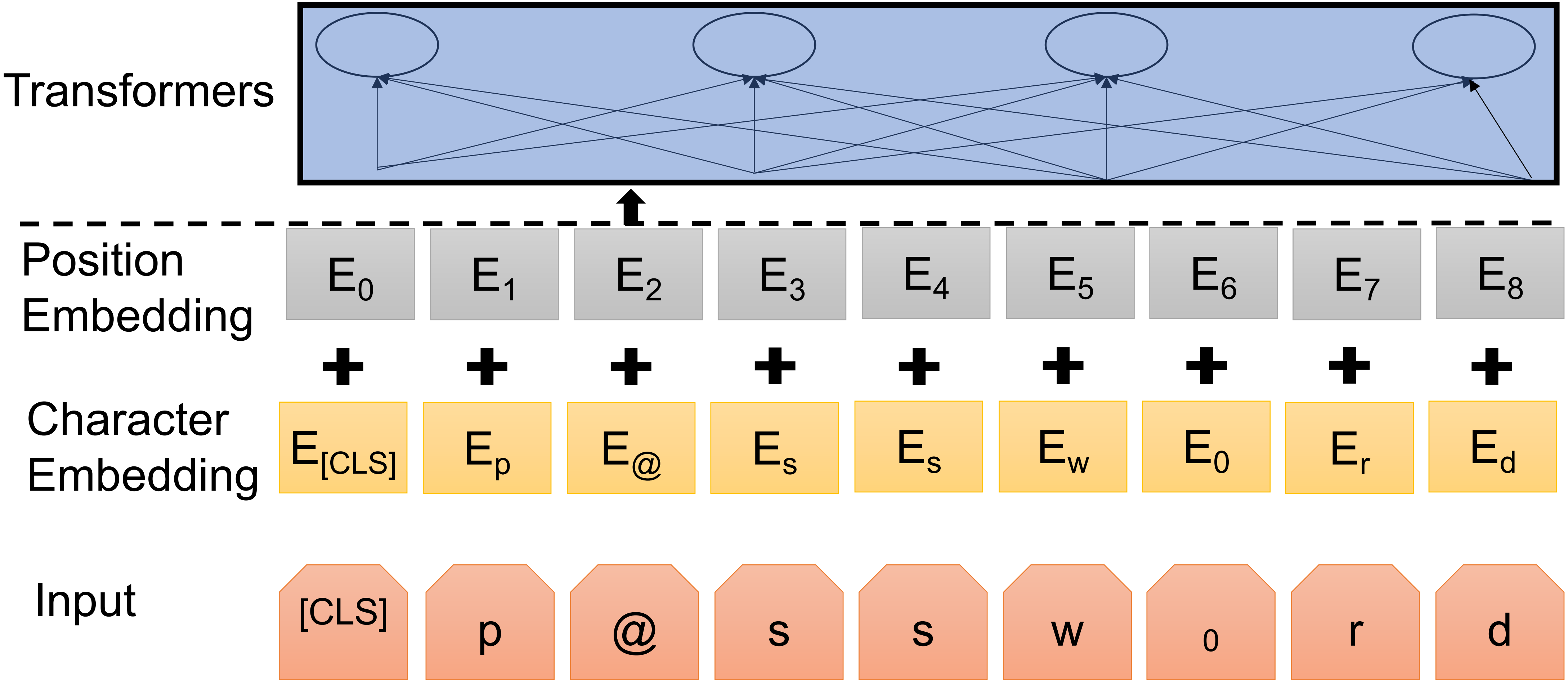
Trivially applying the original transformers to password guessing

Consider case-specific design in three guessing models

For example, contrary to the existing works that uses the sequence-to-sequence mechanism, we use the sequence labeling paradigm in TPG

Password Pre-training Frameworks

Pre-train (Objective: **MLM**) ➡ Pre-trained password parameters (contextualized embedding) ➡ Specific attack models



Password Fine-tuning

- **Modify architecture:** accommodates model's output layers
- **Re-train model:** re-train the model with task-specific objectives and labeled datasets.

All parameters are changed!

Layers	Output shape
Input layers	[batch-size, seq-length]
Embedding layers	[batch-size, seq-length,256]
Transformer block	[batch-size, seq-length, 256]
Transformer block	[batch-size, seq-length, 256]
Transformer block	[batch-size, seq-length, 256]
Transformer block	[batch-size, seq-length, 256]
Fully output layers	[batch-size, seq-length,99]
Output layers	[batch-size, seq-length, 99]

Datasets

Pre-training:

Rockyou-2021

Untargeted Guessing Attacks (CPG, ARPG):

Rockyou-2009, 000Webhost, Neopets, Cit0day

Targeted Guessing Attacks (TPG):

BreachCompilation, Collection#1

(Emails, pwds) → Email: pwd_1, $pwd_2 \ldots pwd_n$

Real-world Guessing Attacks

Conditional Password Guessing:

Guessing Scenarios [CWAE, Pasquini et al., SP-2021]

Pivot selecting (**p***w0rd**) : randomly mask characters with **50%** probabilities in a password, and keep only those produced pivots with at least 5 masked symbols and 4 observable characters

Real-world Guessing Attacks

Conditional Password Guessing:

Guessing Scenarios [CWAE, Pasquini et al., SP-2021]

Pivot selecting (**p***w0rd**) : randomly mask characters with **50%** probabilities in a password, and keep only those produced pivots with at least 5 masked symbols and 4 observable characters

Model Design

Keep the model architecture

$$P(pwd \mid pivot) = \prod_{c_i \in pwd,\ mask_i \in pivot} P(c_i \mid mask_i, pivot)$$

Change the masking mechanisms to be consistent with the pivot selecting

Real-world Guessing Attacks

Evaluation (CPG):

- CWAE; *PassBERT; Vanilla BERT; PassBERT

pivots	Neopets (%)				Cit0day (%)			
	CE	**PT*	*VT*	*PT*	*CE*	**PT*	*VT*	*PT*
common	68.62	74.04	77.25	**80.02**	67.65	75.66	79.90	**83.23**
uncommon	77.35	73.88	79.40	**83.51**	69.30	72.80	76.18	**80.06**
rare	70.62	75.52	76.07	**79.72**	63.70	70.08	71.83	**76.48**
super-rare	69.86	59.51	62.25	**73.41**	45.90	46.11	47.86	**52.50**
average	71.61	70.73	73.74	**79.16**	61.64	66.16	68.94	**73.06**

- Improving the cracking efficiencies significantly.
- Password pre-training can provide notable improvement.

Real-world Guessing Attacks

Targeted Password Guessing:

Guessing Scenarios [Pass2path, Pal, et, al., SP-2019]

U $\xrightarrow{\text{Leaks}}$ Passwords: Alice1997 ↷ A $\xrightarrow{\text{Generates}}$ Password variants: @lice197!

Real-world Guessing Attacks

Targeted Password Guessing:

Guessing Scenarios [Pass2path, Pal, et, al., SP-2019]

U Leaks Passwords Alice1997 → A Generates Password variants @lice197!

Model Design

A l i c e 1 9 9 7

(rep,a) k k k k k (del) k (rep, 7!)

Predict the edit operations, i.e., we pre-defined [(replace, !) , keep, (delete, null) , (replace, 7!)], for every character.

Real-world Guessing Attacks

Evaluation (TPG):

- Pass2path; *PassBERT; Vanilla BERT; PassBERT

Attack model	BreachCompilation (%)			Collection#1 (%)		
	10	100	1,000	10	100	1,000
Pass2path	6.42	11.52	14.71	4.37	10.84	14.98
*PassBERT	12.63	15.67	17.94	11.21	15.42	18.22
Vanilla BERT	**12.72**	**15.79**	**18.01**	**11.35**	15.45	**18.23**
PassBERT	12.68	15.71	17.96	11.24	**15.47**	18.21

- Improving the cracking efficiencies significantly.
- Password Pre-training can provide marginal efficiency improvement.

Real-world Guessing Attacks

Adaptive Rule-based Password Guessing:

Guessing Scenarios [ADaMs, Pasquini et al.,USENIX-2021]

R

- All rules [(a →@), (delete last three characters), (add 123 to the end)] to a word (password), e.g., Hashcat
- Adaptive rules [(a→@)] to a word

Real-world Guessing Attacks

Adaptive Rule-based Password Guessing:

Guessing Scenarios [ADaMs, Pasquini et al.,USENIX-2021]

R

- All rules [(a →@), (delete last three characters), (add 123 to the end)] to a word (password), e.g., Hashcat
- Adaptive rules [(a→@)] to a word

Model Design

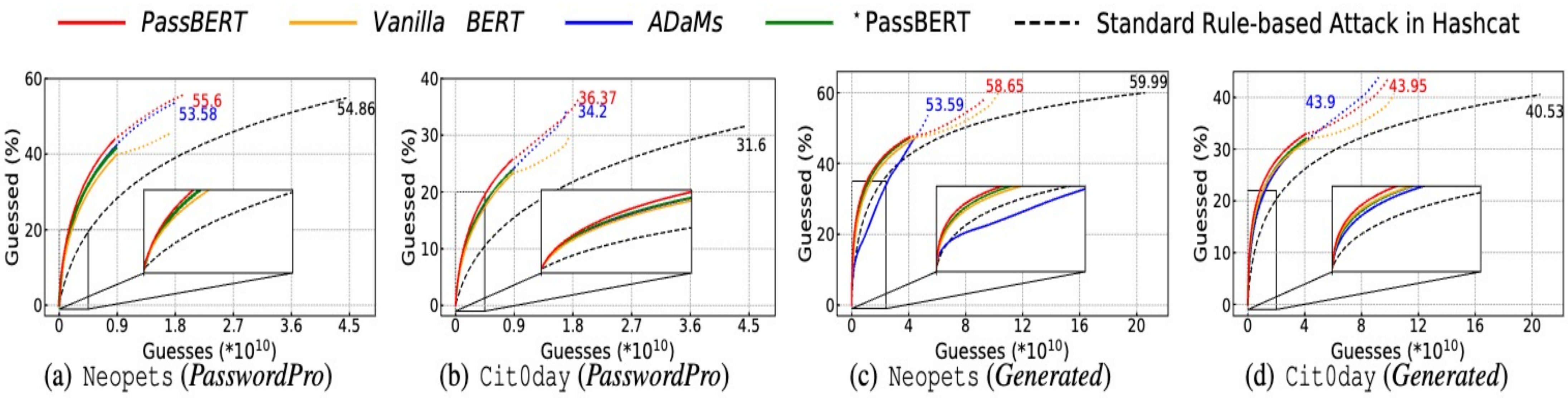
Calculate the probability between a word and a rule

$$< P(w, r_1 \in \mathscr{R}), P(w, r_2 \in \mathscr{R}), \ldots, P(w, r_{|\mathscr{R}|} \in \mathscr{R}) >$$

Regard the rules with larger probability threshold as adaptive rules

Real-world Guessing Attacks

Evaluation (ARPG):



- By employing password pre-training, PassBERT outperforms ADaMs, leading to improved cracking efficiencies.
- ARPG demonstrates comparable cracking rates to final efficiencies in standard rule-based attacks in Hashcat within the top 20% guesses.

Pre-training Effects

- Pre-training can yield notable improvements in untargeted guessing attacks, while only providing marginal improvements in targeted guessing attacks.

- It is necessary to have a pre-trained password model, which can provide notable gains in untargeted guessing scenarios.

Takeaways

- We demonstrate the potential threat from real-world guessing attacks (e.g., CPG, TPG and ARPG), which can significantly threaten password-based authentications.

- The advanced attacks lead to valuable ideas in the design of PSMs, and push PSM towards comprehensive strength evaluation like hybrid password strength meters.

character strength level:	p @ s s w 0 r d 1 2 3
potential risks from target guessing attacks:	The input of "p@ssw0rd123" can be cracked when trying 825 guesses given the leaked "p@ssw0rd"; make it more complex!

- Pre-training on an unsupervised task (e.g., MLM), either upon the web corpus or the passwords, are generally beneficial to guessing attacks in the password domain.

Thanks !