

PasswordNinja: Password Composition and Complexity Insights from Manually Labelled Datasets

Sirvan Almasi and William J. Knottenbelt
Imperial College London

Abstract

This paper presents the first extensive, manually labelled password dataset, encompassing the `hotmail` dataset and a portion of the `phpbb` dataset, decomposed into chunks, words, structures, tags, and transformations. Traditional modelling of password strings, either character-by-character or in chunks, overlooks the nuanced details of passwords. Our approach models passwords through a series of transformations and concatenations, with words or names as the foundational elements. This dataset reveals the relative resilience of different structures and facilitates an evaluation of password composition policies. Because password labelling is a human resource intensive task, we explore the efficacy of human agents and language models in automating this process. Our research utilises language models to expedite password labelling. By training various models on the labelled `hotmail` and `phpbb` dataset, we prove the practicality and effectiveness of language models in decomposing password structures compared to human agents.

The dataset is publicly available at <https://anonymous.open.science/r/passwordninja-2262>.

1 Introduction

The lack of consensus on a practical password composition policy (PCP) stems from our inadequate understanding of password compositions and, ultimately, what constitutes a strong password. If any best practices on password composition exist, they are often ignored [1]. PCPs are designed to nudge users toward creating strong passwords; however, these policies have rendered the most usable form of authentication [2], the password, less user-friendly. Our journey to discover a usable PCP, and thereby an understanding of what makes a password strong, begins with the first large-scale, human-labelled dataset—the `hotmail` [3] passwords leak. This dataset will be useful not only for password composition policy analysis but also for password modelling and examining semantic and structural aspects.

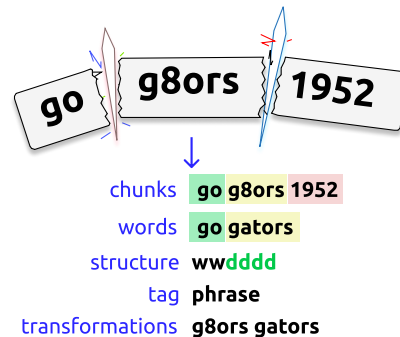


Figure 1: Example of the labelling process. The password `gog8ors1952` is sliced into core constituents, then the meaningful words are extracted. Such a labelling process can capture details such as transformations.

The current analysis of password strings is confined to high-level assessments. Traditionally, studies have focused on syntax and general patterns, such as length, digit count, and dictionary search. However, these approaches have proven to be both inaccurate and inconsistent across different research [4]. The field demands repeatability facilitated by superior datasets. Without access to a decomposed and labelled dataset, evaluating password guessing methods beyond simple success rates remains challenging. The absence of a detailed dataset impedes further exploration into the effectiveness of diverse approaches. The results of our manually labelled dataset in Section 4 demonstrates the utility of a labelled dataset.

Understanding password composition is critical in password modelling, as it significantly impacts the precision and efficacy of predictive models. Traditional methods, such as [5], typically analyse passwords at the character level, considering the likelihood of each character’s occurrence in sequence. More sophisticated techniques, like chunking [6], break down passwords into larger, meaningful segments rather than isolated characters, offering a refined perspective on password composition. Nonetheless, these approaches do not provide the detailed insight needed to fully understand nuances such

as word transformations, semantic contexts, and structural patterns within passwords.

We aim to address the above limitations through a closer analysis of the password strings themselves. The thesis of this experiment is that the core of a password is one or a set of ordinary words, such as a dictionary word or a name. These word(s) are either transformed or concatenated with other variable text or digits to meet a password policy or simply make it more complex.

Data labelling is highly time-consuming and expensive. Thus, we asked ourselves: How can we expedite the labelling process? We can hire more human agents or develop algorithms and models for assistance. We tested both approaches, employing paid freelancers and utilising language models, to aid in the labelling process (see Section 6).

This paper presents **three contributions** with the aim of advancing the security of human-chosen secrets—passwords.

1. **Human-labelled password corpus** (§4): We decompose and label the `hotmail` password strings (8 292) into chunks, words, structure, tags, and transformations, where applicable. To our knowledge, this is the first dataset of its kind. Insights derived from our labelled dataset are discussed, and the dataset is made available for academic research¹.
2. **Password complexity** (§5). Utilising our labelled dataset, we evaluate secure and practical password composition strategies. We find that password structures, such as four-word passphrases without semantic links between words, offer both security and practicality.
3. **Large Language Models and human labellers** (§6) Labelling data is time-consuming and costly, prompting exploration of alternatives to expedite the process. We engaged freelancers from Fiverr [7] and assessed their work against Large Language Models in terms of accuracy, cost, and time. The preliminary results indicate that language models are a promising tool for further experimentation.

2 Background

Password complexity. Password strength measures the difficulty of guessing or cracking a password. However, there is no consensus on how to quantify this difficulty. Nonetheless, some lower-bound measures are widely accepted, such as the number of characters.

$$H(X) = - \sum_{x \in \mathcal{X}} p(x) \log p(x) \quad (1)$$

Shannon’s entropy (Eq. 1) quantifies the uncertainty or information content associated with a random variable. In the context of password strength, it measures how unpredictable

or "random" a password is. The higher the entropy, the more difficult the password is to guess, making it stronger. This metric estimates the minimum number of bits required to encode the password. However, the entropy measure for passwords is often given in the form of Eq. 2.

$$E = \log_2(R^L) \quad (2)$$

R represents the total number of characters in the character sets used in the password, and L is the length of the password.

The limitation of Eq. 2 is that it assumes all R characters are equally likely, disregarding the dependency between characters. This approach oversimplifies, as passwords like `8zm!U6gNL%` and `p4$sw0rd1!` are calculated to have the same entropy. However, such an assumption is unrealistic, as the latter password is essentially the word "password" with predictable transformations and common end-character additions.

Password Composition Policies (PCP). The *National Cyber Security Centre (NCSC)* [8] in the UK and the *National Institute of Standards and Technology (NIST)* [9] in the USA are leading authorities in cyber-security. Their recommendations, particularly on password policies, should serve as standards or benchmarks in the field. Specifically, the NCSC advocates for the use of three random words [10] and advises against mandatory password complexity requirements [11]. This approach, detailed in [12], is based on the principle that a sequence of three random words is both easier for users to remember and sufficiently complex to deter unauthorised access, striking a balance between security and user convenience.

- **Length.** Three words will likely meet the minimum length requirements.
- **Impact.** Easy to adopt and understand.
- **Novelty.** It is unlikely a $3 \times$ words password will be in an existing database.
- **Usability.** More likely to be remembered.

NIST’s [13] PCP is as follows. Passwords should have a minimum length of 8 characters, but systems should support more than 64 characters. It is advised to compare the selected password against prior breach data, dictionary entries, and repetitive patterns. Context-specific terms should also be considered. Additionally, users should be aided with strength meters to gauge password robustness. While rate limiting should be enforced during authentication attempts, the imposition of specific composition rules is discouraged.

3 Method

This is a method to calculate the pre-image of a password through decomposition. This task involved collecting and

¹<https://anonymous.4open.science/r/passwordninja-2262>

cleaning leaked password datasets then manually labelling the individual passwords to decompose into chunks, words, structure, and consequently tagging them. Such a detailed decomposition has been lacking in the field, where academics have crudely modelled password strings based on individual characters and chunks. The newly created labelled dataset then was used to train various machine learning models to do future password decomposition with little human intervention. In the process we experiment with methods in aiding us in the labelling process (§6).

Throughout the paper we refer to keywords such as semantics and structure. Their use and definition is as follows.

- **Lexical Analysis:** Identifying the basic components such as letters, numbers, and special characters.
- **Semantic Decomposition:** Breaking down the password into meaningful units or words (if any). For instance, identifying that "cat" and "dog" are two words in the password "catdog123".
- **Pattern Recognition:** Identifying common patterns such as repeated characters, sequential numbers, and common substitutions (like using "4" for "A" or "3" for "E").
- **Structural Analysis:** Identifying the structure of the password, such as the order of different types of characters (e.g., letters followed by numbers, or interspersed special characters).
- **Transformation Analysis:** Identifying any transformation applied to parts of the password, like leetspeak substitutions or capitalisation patterns.

Data We began by compiling a frequently utilised dataset list from prior studies. Subsequently, we expanded this list with open-source materials sourced from forums. The datasets previously employed are open-source and readily accessible. We have concentrated on datasets predominantly in English, Spanish, and German. We selected the `hotmail` dataset due to its substantial size of 8 929 entries, the phishing method used for its breach, and its bilingual content in English and Spanish. Common datasets such as `Rockyou`, `phpbb`, and `000webhost` also emerged; however, their vast sizes exceed our capacity for manual labelling given the limited human resources available.

The `hotmail` dataset, which leaked in 2009 [3], comprises 8 929 records of passwords. Public sources indicate that these passwords were procured through phishing. Abundant evidence within the dataset supports this claim. For instance, we observe numerous passwords with inverted capitalisation and similar passwords that contain typos, suggesting that users might have unintentionally activated `CAPS LOCK`.

Labelling We segmented the 8 929 leaked passwords based on their structural characteristics. Numeric-only passwords were excluded. The remainder was categorised into three groups: 1) Passwords following an $L_n D_n$ pattern, consisting of letters followed by digits; 2) Passwords with an L_n structure,

Label	Description
R	Random passwords with no apparent structure.
R2	The password has a distinct non-random structure but we cannot decompose it into meaningful components.
R3	The password has a distinct repeating pattern such as <code>abcd123</code>

Table 1: Labels for passwords that have a degree of randomness to them or the passwords are undecipherable to the human labeller.

Ser	Tag	Description
1	word	Standard dictionary word or common.
2	name	Given real or fictional names.
3	phrase	Password is composed of multiple words.
4	location	Places such as towns, cities and countries.
5	date	If majority of the password is made up of a date.
6	org	Organisation
7	object	Specific objects such as car models or artefacts.
8	email	Email address like passwords.
9	website	Entire string composed as a web URL
10	kp	Keyboard pattern, e.g. <code>qwerty</code>

Table 2: Keywords used to tag and identify passwords based on their composition.

including letters and symbols; and 3) Those that did not fit the previous categories, labelled as $!L_n D_n$. We then employed the Markov n -gram entropy method to sort the passwords. This process involved semantically breaking down passwords into identifiable segments, extracting meaningful words, and then labelling the password structure based on these components. Each password received a tag reflective of its primary element—word, name, or phrase, for example—outlined in Table 2. Passwords lacking overt semantic content were designated as R2, denoting a human-generated structure that was ambiguous to the labeller. Passwords identified as machine-generated or random received an R1 tag, while those with repetitive structures or keyboard patterns were marked as R3. We have summarised these classifications in Table 1.

Training In our experiment evaluating the effectiveness of different agents (both human and machine) in labelling password strings, we fine-tuned several Large Language Models (LLMs). The GPT-3 models were fine-tuned via the OpenAI API. The LLama2 and Mistral-7B models underwent fine-tuning with the QLoRa [14] method on a machine equipped with an AMD EPYC 7b13 64-core CPU, 256GB RAM, and two RTX 6000 Ada GPUs (96GB total VRAM).

4 Results & Evaluations

The labelled `hotmail` dataset is our core contribution and this section presents an overview and an initial analysis of it. We manually decomposed and labelled the entire `hotmail`

pwd	chunks	words	structure	tags	transformations
caracteristicas10	caracteristicas 10	caracteristicas	wdd	word	
estatica4	estatica 4	estatica	wd	word	
manterola84	manterola 84	manterola	ndd	name	
onstantinopla84	onstantinopla 84	constantinopla	wdd	location	onstantinopla constantinopla
elingeniero20	el ingeniero 20	el ingeniero	wwdd	word	
centella2	centella 2	centella	wd	word	
indiferencia8	indiferencia 8	indiferencia	wd	word	
veinti7	veinti 7	veinti	wd	word	
tere50	tere 50	tere	wdd	word	
Bertita1	Bertita 1	bertita	nd	name	
alohanene2	aloha nene 2	aloha nene	nnd	name	
mialiasessANGELO1	mi aliases ANGELO 1	mi aliases angelo	wwnd	phrase	
andreateamo3	andrea te amo 3	andrea te amo	nwwd	phrase	

Table 3: Table showing a small subset of the labelled *hotmail* dataset. We decomposed the passwords from the *hotmail* dataset into chunks, words, structure, tags, and transformations (if applicable). Such decomposition allows for better password modelling and nuanced understanding of human behaviour when creating their chosen passwords.

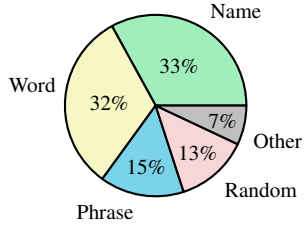


Figure 2: Composition of **tags** (*hotmail* dataset) as a percentage. The core constituents of majority of passwords are words, names, or phrases—this supports our thesis in this paper. The total number of passwords is 7 235, and *other* is composed of 4%: location, 2%: organisation, and 1%: date.

dataset, comprising 8 929 records. Specifically, we labelled 7 235 records that were not solely numeric. Additionally, we labelled 2 000 records from the *phpbb* dataset for comparative analysis and further experiments (discussed in subsequent sections). Table 3 provides a snapshot of the labelled dataset. Given that we labelled the dataset into *chunks*, *words*, *structure*, *tags* and *transformations*, we will analyse each of these components separately in this section. This section will primarily focus on the *hotmail* dataset as it is completely labelled and therefore can be used to draw more useful insights.

The time it takes to label a password varies due to their complexity, but we estimate approximately 170 hours were spent labelling the *hotmail* and a portion of the *phpbb* dataset. A single labeller would likely need 35 to 45 days for such a task. In a subsequent experiment, two freelancers took about 5 days each to label 2 000 records.

4.1 Words

Here, words refers to a column in our dataset and it contains extracted (meaningful) information from the sliced passwords, such as words, names, locations, organisation names, and etc. If a word is transformed in the original password string then we would transform it back to the original form, e.g. *g8ors* → *gators*. We observed 5 749 unique *words*. This rich source of data opens up a vast field of experimentation and new perspectives to analyse passwords from. So, we begin with the frequency of the words in the unique password strings. Figure 8 (in Appendix) shows the most frequent words in the labelled dataset.

With this data we can analyse the distribution of the words. An interesting hypothesis is that passwords in a dataset follow a power law distribution, studies such as [15] present supporting evidence for this. The extracted words in the unique passwords also exhibit a power law similar to Zipf’s law. This can be seen in the Figure 3 where the log-log plot of word frequency and rank is linear: the frequency of the n^{th} ranked word $f_n \propto n^{-\alpha}$ [16].

4.2 Structures

Structure of the password indicates the order and position of different types of characters, digits and words. Table 5 shows the most common structures for the *hotmail* dataset. Majority of the passwords are composed of words or combination of words (phrases). The most common structure is a *w*, which is prone to simple dictionary attacks.

Our method of labelling the structures is certainly up for debate, however, they are flexible and can be converted. For example, for the password *mialiasessANGELO1* has the structure of *wwnd*. This structure can be converted to one presented in the PCFG [17] model [18] where the authors use

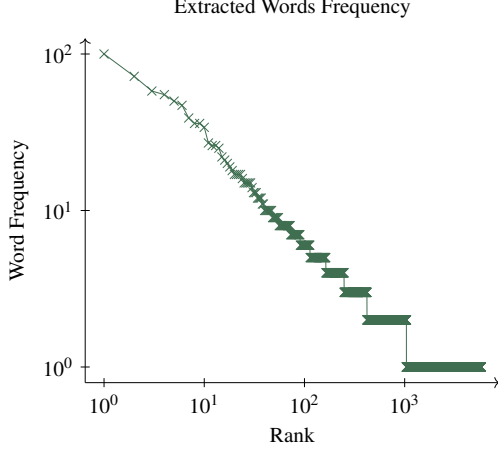


Figure 3: Plot of Rank v.s. Word Frequency (log-log scale) for the `hotmail` dataset. Words are extracted from the unique password strings in the dataset. The distribution exhibit a *power law* similar to Zipf’s law.

L_n , D_n and S_n to represent alpha, digit and special variables (n represents the count of the specific variable). We can extend these to include W_n and N_n for **word** and **name** variables. So, for the example password (structure: `wwnddd`) can be represented as $W_2N_1D_1$. So, one may be able to conduct a PCFG [17] style attack with the knowledge of these structures.

The distribution of the structures exhibit a power law—similar to Zipf’s law if we plot the frequency of the structures against their sorted Rank (as shown in Fig. 4). This can support our thesis that the core of the password is a word or a set of words (a phrase). Curious question for future work is whether other datasets follow a similar power law as seen in the `hotmail` dataset.

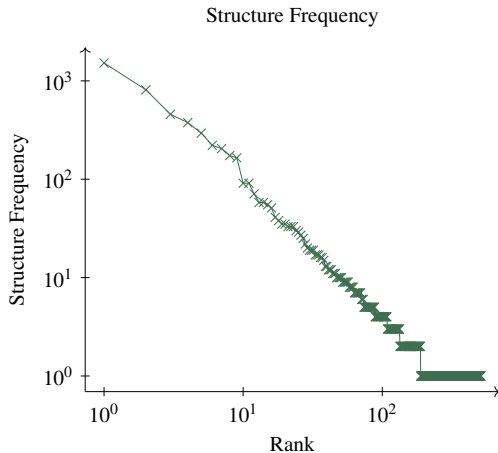


Figure 4: Plot of Rank v.s. Structure Frequency (log-log scale) for the `hotmail` dataset. The distribution exhibits a *power law* similar to Zipf’s law.

4.3 Tags

The tagging of passwords in our study enables a refined analysis of their principal components, identifying whether they consist of names, words, locations, etc. This categorisation facilitates the tailoring of attack dictionaries to specific password compositions. Future research could explore if these compositional trends are consistent across other datasets.

Our analysis reveals that the majority of passwords are composed of words and names, as depicted in Figure 2. These are followed by phrases and passwords classified as Random, which we further divided into R1, R2, and R3 categories. Among the 7 235 records analysed, 946 were labelled as R2, indicating a substantial proportion of passwords appear random but are vulnerable to brute-force attacks. The distribution of the remaining random passwords is as follows: 23 classified as R3 and only one as R1. Additionally, 35 passwords were identified as keyboard patterns.

4.4 Transformations

We observed 324 transformations in the `hotmail` dataset. Our partial labelling of the `phpbb` dataset revealed even more intriguing word transformations. Identifying the exact nature of a transformation can be challenging, especially for non-native speakers of the language used in the passwords, leading to less than 100% confidence in our identifications. Transformations with nearly zero confidence were labelled as R2.

Contextual information can also reveal more information about the transformation, for example: the string `01well` can simply be *zero*, *one* and the word "well", but if we add `9801well` or `198401well` then it becomes apparent that it is "orwell" (referencing the book *Nineteen Eighty-Four* [19] by George Orwell). This example is a clear illustration of why word search and algorithmic means of labelling passwords are difficult. Such nuanced patterns make labelling difficult for a human who cannot make the connection between `84` and `01well` or simply doesn’t have the knowledge of that specific book and author.

Transformations can also encompass multiple words, such as mingling them or altering their order. Such transformations are less frequent than those at the word level. Hence, we categorise these as *phrase-level* transformations, detailed with relevant examples below.

Phrase level transformations:

- **Hybridisation**

- **Overlapping letters:** Such as `r3al0v3` → *real love*, where numbers are used to represent letters that look similar, overlapping in meaning.
- **other** More complex mixing examples: `castilleo` → *leo castillo*: A rearrangement of the letters to form the correct name.

- **Change of order.** Changing the order of the words or chunks in the password.
pedro te amo \rightarrow te amo fred.
- **Repetition.** e.g. memememe

Word level transformations:

- **Homophones.** These are words that sound similar but have different meaning, e.g. 2 \rightarrow to
- **1337 speak** is the replacement of words with numbers and letters to create alternative spellings that resemble leet or 'elite' speak. This replacement would either make the word visually look or sound similar. Examples are shown in the text below.
 - **Visual.** 9801well \rightarrow 1984 Orwell: Numbers are used to resemble the visual appearance of the corresponding letters.
 - **Sound.** articul8 \rightarrow articulate: The number '8' sounds like 'ate', making the word sound like 'articulate'.
 - **Sound.** 2rus \rightarrow trust: The number '2' represents 'to' and the reversal of 'rus' to 'sur' makes the word 'trust'.
 - **Sound.** pahswrd \rightarrow password: Phonetic spelling that omits certain letters but still resembles the sound of 'password'.
- **Slang.** Shorthand or informal language.
 - gurlz \rightarrow girls
 - luvme \rightarrow love me: 'luv' is a common slang for 'love'.
 - skure \rightarrow secure: Phonetic spelling of 'secure'.
- **Abbreviation/Elision.** nbtwin \rightarrow inbetween: Shorthand for 'in between'.
- **Reversal.** Flipping the letters around, as in drowssap \rightarrow password.
- **Syllable swap.** Mixing up the order of syllables or parts of the word, like wordpass or word\$pass.
- **+/- characters.** Omitting one or more characters from a word, such as bloo \rightarrow bloom or asasin \rightarrow assassin.
- **Elongation.** Extending one or more characters in a word, like psssss or sammmmm, often to convey a drawn-out pronunciation.
- **Misspelling:** Intentional or accidental incorrect spelling of words.
- **Doubling letters.** Repeating one or more letters, as in biiaankaa \rightarrow bianka.
- **Random L or D insertion.** Inserting letters at random points, such as cancerols \rightarrow cancerous or plu0lto \rightarrow pluto. Often they occur between syllables in a word.

One can also relate these transformations to the rule set in Hashcat [20].

Adopting a more abstract methodology, we can represent the process as follows: Let \mathbf{W} denote the initial word prior to any transformations. We introduce a set of transformation functions $\{f_1, f_2, f_3, \dots, f_n\}$, where each function f_i implements a specific alteration to a word. For instance, $f_i(\mathbf{W})$ might symbolize the *syllable-swap* of \mathbf{W} if f_i designates the *syllable-swap* function. These functions can be sequenced to execute more intricate transformations, such as $f_j(f_k(\text{password}))$ potentially altering the input to word\$pass if f_j and f_k correspond to *syllable swap* and *random insertion*, respectively.

In the context of a probabilistic transformation system, let $P(f_j|f_i)$ denote the likelihood of applying f_j subsequent to f_i . This signifies the chance that f_j is the ensuing transformation given that the previous one was f_i . Assuming the steps are independent, the total probability of such a sequence materializing can be expressed as the product of the individual probabilities.

5 Password Complexity

In this section, we will demonstrate why the NCSC's [8] password recommendation of *at least three random words* stands as the most practical advice to date. Analysing passwords inevitably leads to considerations on how to forge secure passwords. The security level of a password, gauged through various strength or complexity metrics, is essentially governed by specific *policies*. The discourse around password policy and complexity is extensive, with some asserting that passwords which are both human-chosen and memorable cannot be secure [18]. Here, we will examine these assertions using our labelled dataset as evidence.

5.1 Attack Model

We assume a scenario where the attacker has access to the hash of the target's password. Therefore, it is an offline type of attack and the attacker is not limited to the number of guesses they can make. The methods of attack deployed by the attacker, or the tools they use is inspired by what happens in practice. An example of such practice can be taken from the post-challenge report of Team Hashcat: Every year, KoreLogic [21] hosts a password cracking challenge at the DEF CON conference [22]. Team Hashcat (the developers of the Hashcat [23] password guessing software), have been the winners for the last 3 years. Their 2023 lineup consisted of 20 hackers, 47 FPGA boards, 78 GPUs and cloud infrastructure. Their post-challenge report [24] is a good illustration of an attack strategy, especially given that the 2023's challenge was emulating a real world scenario. None of the prominent machine learning techniques were used except PCFG [17] in order to crack passphrases (2 to 4 words), which was trained on the existing cracked passphrases. Much of the manual

work revolved around deciphering the underlying patterns in order to reduce the search space.

5.2 Insights on Password Strength

Our analysis reveals two key insights into password security. Firstly, users tend to select words from a non-uniform distribution as shown in §4.1, significantly narrowing the search space and making passwords more predictable. This observation underscores the vulnerability of common password choices to guessing attacks. Secondly, the inherent structure of passwords—particularly those most commonly used—reveals considerable information, facilitating attacks like Probabilistic Context-Free Grammar (PCFG [17]). As detailed in Table 5, the predominant password structures are **w** (word) and **n** (name), both of which are highly susceptible to dictionary attacks. Furthermore, the efficacy of these attacks varies with the hash function used. To illustrate, the table includes the time required to execute dictionary or hybrid attacks using consumer-level hardware across different hash functions, highlighting the practical implications of these security insights.

At the tail end of Table 5 we see structures, such as **www** (or **w₄**) that are resilient even if they were hashed using MD5. So, passwords comprising multiple words exhibit better security, withstanding hybrid attacks for extended duration, using the same hardware. However, their strength is compromised if the selected words have a high probabilistic correlation. In cases like the pattern "te amo x", where x represents a person's name, predictability increases significantly, as the phrase "te amo" often precedes a name.

To counteract this predictability, an effective strategy involves selecting words with minimal semantic or probabilistic linkage. Such a choice ensures that using an online corpus for predictive modelling yields low efficiency in guessing the subsequent word, thereby enhancing password security.

In a Markov process for word sequences in passwords, the probability of a word sequence $W = \{w_1, w_2, w_3, w_4\}$ (where each w_i represents a word) is given by the product of conditional probabilities:

$$P(W) = P(w_1)P(w_2|w_1)P(w_3|w_2)P(w_4|w_3) \quad (3)$$

For a strong password comprising multiple words, we aim for the condition that each word is independent of its preceding word. In such a case, the conditional probabilities should be equal to the individual probabilities of each word, i.e., $P(w_i|w_{i-1}) = P(w_i)$. Therefore, for a password of 4 words where each word is chosen independently we get Eq. 4.

$$P(W) = P(w_1)P(w_2)P(w_3)P(w_4) \quad (4)$$

In an ideal scenario where words are chosen with no semantic or probabilistic relationship to one another, the Markov probability chain should approach a condition where:

$$P(w_i|w_{i-1}) \approx 0 \quad (5)$$

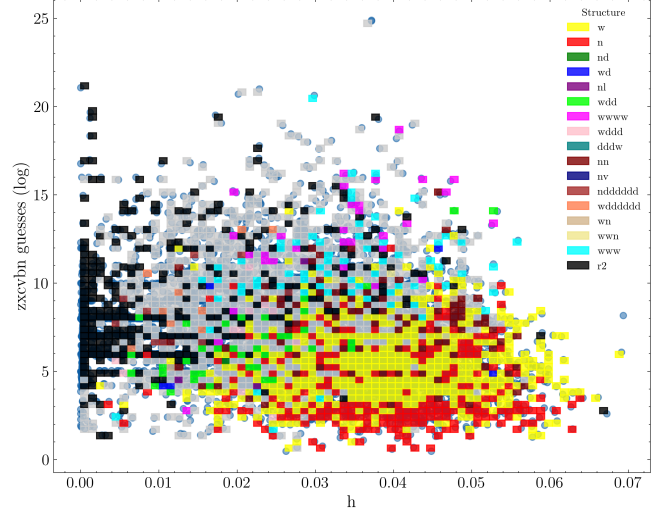


Figure 5: Chart showing the labelled hotmail passwords and their respective structure. The individual grids represent the most frequent structure. On the y axis we have the zxcvbn [25] guesses_log10 metric and on the x axis we have the h = Markov 2-gram scores. As expected, the weakest passwords (**w**: yellow and **n**: red) have a high h score and predicted to be weak by zxcvbn [25].

Hence, for a 4-word password with maximally uncorrelated words, the overall probability of predicting the entire phrase should approach zero.

5.3 Password Composition Policy

Various studies have highlighted the inconsistency of password composition policies and the subsequent disregard for these recommendations [1, 26–28]. Our work will replicate parts of these studies, extending the focus to include government applications. We will discuss the findings with respect to the recommendations provided by NIST [9] and NCSC [8], as mentioned in §2.

Table 6 presents password policies from a range of services, including social media, government, and email providers. Surprisingly, a significant number of these services adopt basic password policies, stipulating a minimum length of just 6 or 8 characters, particularly prevalent among social media and streaming platforms. This simplicity could be attributed to the desire to enhance user experience and boost sign-up rates, as stringent policies might deter potential users. In contrast, government entities, exemplified by the Ealing and RBKC Councils in London, tend to enforce more rigorous standards, possibly due to a lack of concern about user adoption rates. Yahoo’s registration stood out for its rigorous vetting of passwords without specific feedback, only informing users that their “password isn’t strong enough” and suggesting longer en-

Structure		Count	Avg Guesses	Avg Length
zxcvbn score = 4				
r2	r^2	160	12.6	14
www	w^3	57	12.1	14
ww	w^2	48	12.0	14
wwww	w^4	34	13.1	15
w	w	25	11.3	12
nn	n^2	24	11.2	13
wdddd	wd^4	19	11.6	12
wwdddd	w^2d^4	16	11.3	14
nndd	n^2d^2	15	11.5	13
ndddd	nd^4	14	10.8	12
zxcvbn score = 3				
r2	r^2	185	8.9	10
w	w	100	8.8	10
ww	w^2	86	8.8	11
nn	n^2	77	8.8	11
www	w^3	62	9.1	11
wdddd	wd^4	44	8.8	11
ndddd	nd^4	43	8.7	10
wdd	wd^2	38	9.0	10
wwdd	w^2d^2	31	9.1	11
n	n	26	8.9	10

Table 4: Top 10 password structures for each zxcvbn [25] score of 3 and 4. These are the password structures that are categorised in the most secure bin (4) by the zxcvbn [25] algorithm; Avg Guesses is the zxcvbn [25] `guesses_log10`.

tries. This verification was server-based and executed through a POST request. Outlook, interestingly, imposed the most complex requirements, demanding a blend of characters, numbers, and symbols. Github introduced a flexible yet secure approach with two distinct policies: a minimum of 8 characters with at least one digit or a minimum of 15 characters without obligatory digits or symbols. While Github alerted users to potential password vulnerabilities, it didn’t restrict their choices. UK’s HMRC, reflecting the National Cyber Security Centre’s advice, recommends a password comprising three random words with a minimum of 10 characters. Contrary to the opening abstract statement of [29], few of the websites had a password strength meter (PSM).

Evaluation. Password policies can be classified as continuous, involving algorithms or probabilistic methods, which do not allow for concise user instructions on recommended password structures or word choices. Alternatively, password policies can be discrete, exemplified by the guidelines from NCSC [8] and NIST [9], which are clearly defined. Our analysis, based on the hotmail dataset, supports the effectiveness of the NCSC [8] policy. Furthermore, our examination with zxcvbn [25], as illustrated in Figure 5 and Table 4, indicates that the most secure structures involve multiple words, regardless of the probabilistic relationships between the words in this dataset. However, as we demonstrated using Eq. 5, we should aim for no semantic relationship between the words.

6 Experiment: Human and Language Models as Labellers

Data labelling, particularly for password strings, is a labour intensive task. To expedite this process, we explored the effectiveness and cost-effectiveness of human versus machine labelling. We recruited freelancers from Fiverr [7] to represent the human element. On the machine side, we employed several Large Language Models (LLMs), including GPT-3.5-turbo [49], LLaMA2-13B [50], and 70B, along with the Mistral-7B [51]. This approach allowed us to compare the precision, efficiency, and financial implications of utilising human labour against AI technologies in the task of labelling complex datasets like passwords.

Several researchers have explored the application of Large Language Models (LLMs) for data labelling [52–54] with positive results. Notably, this report [55] cites the accuracy and significant speed up from the language models compared to human labellers.

6.1 Freelance Labelling

We tested the utility of freelancers from Fiverr [7] for password labelling. We took two samples of 1000 passwords from the phpbb dataset; we selected passwords that were of length 6 and greater, $L \geq 6$, and non-numeric but contained at least one digit. We hired two separate freelancers for \$50 each. The freelancers were proficient in the machine learning data labelling process but were non-native English speakers. We would categorise them as non-experts in this domain. We used Google Sheet to share the passwords and monitor the labelling progress. This method proved beneficial as we observed interesting behaviours. We took a hands on approach first by correcting the freelancers at the beginning by checking the first 50 labelled passwords. We had to do a lot of correction.

Interestingly, the second human agent enlisted additional helpers to expedite the labelling process, resulting in errors and inconsistency despite clear instructions. This was evident in the labelling document, where some labellers resorted to copying and pasting from search engine results, prioritising speed over accuracy. In contrast, the first agent focused on accuracy, seeking detailed feedback from us, but exceeded their five-day deadline and showed inconsistency in labelling. Accounting for the hands-on help, the accuracy of the extracted words was $50\% \pm 10\%$; this characteristic was the more accurate compared to chunking, deciphering the structure and tagging the passwords. Ultimately, the output from both agents proved unreliable.

6.2 Language Models

Part 1. We started by fine-tuning the GPT3.5-turbo-0613 model with 825 manually labelled passwords, aiming for JSON-formatted outputs. This fine-tuning was carried out



























Structure	<i>log perm</i> *	Frequency		Hashcat using Nvidia RTX 4090				zxcvbn [25] Mean	
		Count	%	MD5	PBKDF2-sha256	scrypt	bcrypt-sha512	Score	Guesses
w		1520	21.0	~ 0	0.1 s	1 min	4 min	1.3	5.36
n		811	11.2	~ 0	0.1 s	1 min	4 min	1.0	4.72
nd		58	0.8	~ 0	0.6 s	12 min	42 min	1.4	5.68
wd		91	1.3	~ 0	0.6 s	12 min	42 min	1.5	5.97
nl		33	0.5	~ 0	1.5 s	30 min	2 h	1.2	5.09
wdd		294	4.1	~ 0	5.6 s	2 h	7 h	1.8	6.69
ndd		221	3.1	~ 0	5.6 s	2 h	7 h	1.6	6.33
nddd		71	1.0	~ 0	56.4 s	19 h	3 day	1.7	6.37
wddd		91	1.3	~ 0	56.4 s	19 h	3 day	1.9	6.84
wsdd		29	0.4	~ 0	3 min	3 day	9 day	2.0	7.28
ddddw		35	0.5	~ 0	9 min	8 day	29 day	2.4	7.95
wdddd		174	2.4	~ 0	9 min	8 day	29 day	2.3	7.59
ndddd		205	2.8	~ 0	9 min	8 day	29 day	2.0	6.90
nn		376	5.2	1.5 s	8 h	1 yr	4 yr	2.0	6.91
ww		457	6.3	1.5 s	8 h	1 yr	4 yr	2.0	7.04
nw		30	0.4	1.5 s	8 h	1 yr	4 yr	2.2	7.51
nddddd		51	0.7	3.0 s	16 h	2 yr	8 yr	2.5	8.05
wdddd		33	0.5	3.0 s	16 h	2 yr	8 yr	2.8	8.84
wwdd		58	0.8	3 min	33 day	111 yr	402 yr	2.9	8.97
nndd		38	0.5	3 min	33 day	111 yr	402 yr	3.2	9.62
wwddd		33	0.5	4 h	9 yr	1.11×10^4 yr	4.02×10^4 yr	3.3	9.81
nwn		41	0.6	9 day	447 yr	5.56×10^5 yr	2.01×10^6 yr	3.0	8.88
www		165	2.3	9 day	447 yr	5.56×10^5 yr	2.01×10^6 yr	3.0	9.26
nww		35	0.5	9 day	447 yr	5.56×10^5 yr	2.01×10^6 yr	2.7	8.18
nnn		27	0.4	9 day	447 yr	5.56×10^5 yr	2.01×10^6 yr	3.3	9.95
wwwwww		55	0.8	171 day	8.70×10^3 yr	1.08×10^7 yr	3.91×10^7 yr	3.5	11.24

Table 5: The table presents the top 25 labelled password structures from the `hotmail` dataset, accounting for 5 032 (69.5%) of the labelled dataset. Hashcat [23] benchmarks indicate the performance of consumer hardware in a hybrid attack. The dictionary size for *w* (words) and *n* (numbers) is set at 5×10^5 . zxcvbn [25] serves as a password strength meter, with scores ranging from 0 to 4, and guess estimates are log based.

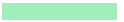

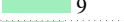




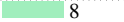





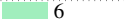



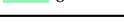

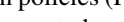
Service	Min length	Capital?	[0-9]	Symbols?	Meter?	Accepted password	Type
github.com (15) [30]	 15	o	o	o	•	github- (x3)	Dev
UK HMRC [31]	 10	o	o	o	o	pa\$\$wordgov	Gov
yahoo.co.uk [32]	 9	o	o	o	o	threeman1	Email
Protonmail [33]	 8	o	o	o	o	passwordman	Email
Reddit [34]	 8	o	o	o	o	passwordman	Social media
X/twitter [35]	 8	o	o	o	o	passwordm	Social media
Microsoft/Outlook [36]	 8	•	•	•	o	Password!1	Email
Spotify [37]	 8	o	o	o	o	passwordm	Streaming
roblox.com [38]	 8	o	o	o	o	passwordm	Kids Game
ebay.co.uk [39]	 8	o	•	o	o	pa\$\$wordm	Shopping
github.com 8 [30]	 8	o	•	o	•	password1	Dev
Google/gmail [40]	 8	o	o	o	o	passwordm	Email
Ealing Council [41]	 8	•	•	•	o	Password1	Gov
RBKC Council [42]	 8	•	•	•	o	Password1!	Gov
NHS [43]	 8	•	o	o	o	Password	Gov
Facebook [44]	 6	o	o	o	o	qgizhac	Social media
Instagram [45]	 6	o	o	o	o	passwo	Social media
LinkedIn [46]	 6	o	o	o	o	passwo or 123456	Social media
Netflix [47]	 6	o	o	o	o	passwo	Streaming
Amazon.co.uk [48]	 6	o	o	o	o	passwo	Shopping

Table 6: Password composition policies (PCP) for popular websites, local and central government applications in United Kingdom. No clear pattern on the adherence to best practices or recommendations from national agencies.

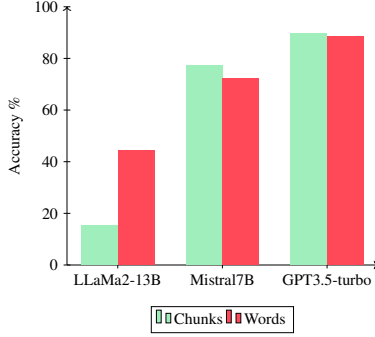


Figure 6: Accuracy of three models in chunking and extracting words from a password string. The models are fine-tuned using 825 passwords from the `phpbb` dataset, and are tested on a separate 300 passwords from the same dataset.

using the OpenAI’s Python API. The fine-tuning cost was $\sim \$5$ and completed in few hours. Testing involved evaluating this model against 300 passwords from the same `phpbb` distribution, distinct from the training set. This initial fine-tuning, despite using limited data, showed success; however, 41 of the 300 (13.7%) passwords required relabelling. The model struggled against transformations, especially elongated passwords such as "ssssue", "6etter", and "vic20oria; the latter we assume to be "victoria". Tokenization is an important aspect of language models. We believe mislabelling could come from tokenisation as well. Doing further tests on the tokenization method was however beyond the scope of this paper.

Encouraged by promising initial results, we conducted further experiments with additional open-source models. We started with smaller models, LLaMA2-13B and Mistral7B, and fine-tuned them using LoRa [56] and QLoRa [57] techniques. The outcomes of the initial experiment, using an 825 labelled training set and a 300-item test set from `phpbb`, are presented in Figure 6.

Part 2. In the second phase of our experiment, we augmented our training set by adding 900 passwords from the `hotmail` dataset. Building on the initial work with GPT-3.5 and Mistral, we expanded our test set to include 3 062 labelled `hotmail` passwords. These passwords consisted primarily of characters (without digits) and were not found in dictionaries, indicating they were either phrases or obscure words. We replaced the LLaMa-13B model with a larger LLaMa2 variant, specifically using the *Riiid/sheep-duck-llama-2*² [58–60] model, which benefits from training on a more extensive dataset compared to the standard 70B parameter model.

²<https://huggingface.co/Riiid/sheep-duck-llama-2-70b-v1>.

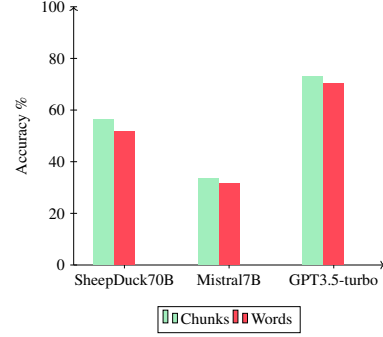


Figure 7: Chart showing the accuracy of three models in chunking and extracting words from a password string. The models are fine-tuned using 1 725 manually labelled passwords from the `phpbb` and `hotmail` dataset. Test data is 3062 passwords from the `hotmail` dataset.

6.3 Evaluation

The models do not generalise well with different structures from ones trained on, e.g. `rosa38` and `38rosa` have the reverse grammatical structures. The labelling of the *structure* were predominantly correct except for the digit counts. The models also found reversing the structure difficult. This $A \rightarrow B$ and $B \rightarrow A$ phenomena in LLM is well documented [61].

The model does not generalise well against languages. We tested the `hotmail` passwords that contained Spanish words with models that were trained on the `phpbb` data and often the models attempted to translate words back into English.

All the models produced well structured JSON data regardless of the amount of data it was fine-tuned with.

Our initial experiment on using language models to label passwords were successful enough to warrant future research in our opinion. Further experimentation of language model is beyond the scope of this paper and there is significant research avenues to explore. Conducting more experiments with different prompting techniques, synthetic data, more human labelled datasets, and different models can prove beneficial.

7 Discussion

The method used to capture a dataset introduces a bias. Often, leaked datasets consist of hashes that have been cracked by various entities and then redistributed. Passwords that remain uncracked are either omitted or only their hashes are included. This omission, along with the presence of uncracked passwords, can distort the analysis. The value of the `hotmail` data lies in its acquisition through a phishing attack, leading us to assume that it includes passwords of all complexities. In contrast, datasets derived solely from obtained hashes are likely to lack the structures considered difficult to crack, such as sequences like `www`. The `hotmail` dataset does not exhibit signs of strict password policies, such as the requirement for

Llama2-13b	Mistral-7B	GPT3.5	Human	Password
robot cat	robocat	robot	robo cat	roboc4t
avi s	b3avis	beavis	beavis	b3av1s
j f ulton	fulton	jfulton	fulton	jfulton275
falcor	falco	falc0r	falcor	falc0r99
leaves one	tm leaves	leafs	leafs	tmleafs1
russian number one	rus no	to run on two rust no one	trust no one	2rusno1
dr p3pper	dr pepper	dr pepper	dr pepper	DrP3pp3r
mitchell	mitchell	?	mitchell	m1tchell
scooby too	scooby doo	scooby doo	scooby doo	scooby2doo
r2	flat line	flatline	flatline	.flat1ne
jlizard	lizard	jlizard	lizard	jlizard1
for me to know for	i know	for me to know for	for me to know for	4me2kn0w4
better one	etter one	letter one	better one	6etter0ne
R2	use	issue	sue	ssssue1
R2	pin	pin	pin	7247pin

Table 7: Experiment 1: Selected comparison of passwords between Llama-13B, Mistral-7B, and GPT3.5 model, comparison is showing word extraction performance of each model. Passwords are from a set of labelled phpbb passwords. Each of the models were trained on a hand labelled set of 1,000 passwords from the phpbb dataset.

special characters, digits, or mixed character cases.

We have used real passwords that belong to individuals that were phished and consequently tricked into revealing their passwords. This raises few ethical issues: whether this in depth analysis will hurt those users? will it reveal any other secrets? will it identify any individuals? This dataset dates back to 2009 and therefore it is highly unlikely that the same users would have kept their passwords the same even if the identity of the individual could be revealed. Identities are highly unlikely to be revealed through our data. We have not used any PII or any other information that could link back to individuals. This dataset has also been previously used in other research papers.

7.1 Limitations

There is a portion of the dataset that is labelled as R2 that could be studied further. Some of these passwords were difficult to label. The presence of human errors and inconsistencies in the labelling process is a notable limitation of our work. It is reasonable to assume that hand-labelled datasets will contain some inaccuracies. Nevertheless, these errors are generally negligible. To mitigate these issues, we propose the development of an online dashboard that allows users to continuously identify and correct any existing errors.

Labelling passwords in non-native languages for the human agent introduces additional inconsistencies, primarily due to limited vocabulary and unfamiliarity with linguistic nuances. Implementing an online portal where native speakers can rectify these errors would be advantageous over time. This system could also facilitate the creation of an extensive phonetic dictionary, encompassing words and names typically challenging to locate in centralised repositories.

This study’s applicability to datasets in languages other than English or Spanish, such as Chinese, is a notable limi-

tation. While Chinese datasets and literature are extensively researched, direct model transfer is not feasible. However, the methodology and insights into password patterns might be adaptable. Investigating how cultural differences manifest in password choices, including variations in structural patterns, vocabulary categories, and word types, could yield significant insights. Assuming that literature and art reflect their societal and cultural contexts, passwords may similarly serve as indicators within the same domain.

Some passwords are constructed using abbreviations, acronyms, or codes comprehensible only to their creators. Decomposing these into meaningful components is often impractical, warranting their categorisation under the ‘R2’ label. However, these passwords are typically short and susceptible to basic brute-force attacks, diminishing the security of their cryptic elements. Our research indicates that these passwords are frequently combined with other significant words or extended numerical sequences, exemplified by strings like jph928312730 or gke8383.

8 Related Work

We believe our labelled dataset is unique. However, the literature on password syntax, modelling, composition, complexity, and policy is extensive.

Initial password modelling techniques, such as the Markov n-gram model [5], focused on individual characters, positing that “the distribution of letters in easy-to-remember passwords likely mirrors the letter distribution in the user’s native language.” We now validate this hypothesis at the word level. The observation that letter distribution is not uniform holds true for words, exhibiting a power-law distribution akin to Zipf’s law. Modelling passwords based on Probabilistic Context-Free Grammars (PCFG) [18] was the next innovation. Our insights into empirical password structures can refine PCFG

modelling. Incorporating \mathbf{W}_n and \mathbf{N}_n as variables for words and names respectively enhances the computational viability of modelling more intricate structures. Subsequent advancements have leveraged machine learning techniques in password modelling, including neural networks [62], GANs [63], RNNs [64], and Transformers [65], among others.

Historical analyses of password characteristics have focused on *length*, *count*, and *character-level structures* [66–73]. Our research advances this field by delving into more nuanced aspects of passwords with increased precision. Furthermore, the evolution of language models, combined with our successful prototypes, reveals a cost-effective route for labelling additional passwords.

Despite extensive research into password complexity measures and meters, recommendations for password policies remain scarce. Bonk et al. [74] offer guidance on constructing passphrases, effectively extending the NCSC’s [8] three-word suggestion to longer sequences, such as seven words or more. Zxcvbn [25] employs dictionaries and bespoke rules to gauge password strength on a scale from 0 to 4. However, zxcvbn [25] has its limitations; for example, it incorrectly parses the password "gomythsun" as "go myths un", impacting its strength rating. Additionally, it struggles with transformations like "numbr" from "number". Wang et al. [29] delve deeper into evaluating password strength meters. Their endorsement of zxcvbn [25] and Markov n-gram models informed our research approach.

Gerlitz et al. [26] provided a comprehensive analysis of password composition policies in Germany. Our shorter analysis (§5.3) for some UK agencies extended their work. Similar to [1], we also showed the inconsistency in composition policies in common websites.

In the analyses of Chinese passwords [67], the authors conduct a broad analysis on the syntax of passwords. Our research extends [67] by delving deeper into the password structures. For instance, they identified *LLLLLL* as the predominant structure within the *Rockyou* dataset, typically representing a word as we showed in our results. Further structural analysis on various password datasets is presented in [75]. Das et al. [68] explored password syntax, notably introducing analysis on word or phrase transformations, echoing aspects of our research. However, we disagree with their classification of sequential keys, alternate keys, and sequential alphabet as transformations, viewing these more as sequences. Riddle et al. [76] delve into the composition and semantics of passwords, with an emphasis on the psychological significance of words.

In "Investigating the Distribution of Password Choices" [77], the analysis of the *hotmail* dataset concluded with a distribution resembling Zipf’s law. Our analytical efforts advance this observation, demonstrating that extracted words from unique passwords also follow a similar distribution.

The deployment of language models, particularly Large Language Models (LLMs), in this arena is notably scarce. Rando et al. [78] ventured into modelling passwords using

the GPT-2 framework. Considering the broad embeddings encompassing various languages, the semantic breakdown and analysis of passwords stand out as a pioneering application for language models in our view.

9 Conclusion

In this paper, we presented the first large-scale password dataset, which includes over 8 000 passwords from the full *hotmail* dataset and a selected subset of the *phpbb* dataset. This dataset covers a range of password complexity elements such as chunks, words, structures, tags, and transformations. Using this data, we critically assess current password policy recommendations. Our analysis indicates that password configurations, particularly those consisting of four consecutive words (w^4), can be considered secure. We determine that the policy recommending *at least three random words* is the most practical advice so far.

We also investigated methods to streamline the time-consuming and costly process of password labelling. We employed freelancers and compared their performance with that of language models (LLMs) like GPT3.5-turbo, Mistral7B, LLaMa2, and their variations. Freelancers took about 7 days to label 1 000 passwords each, with results that were sometimes inconsistent. On the other hand, language models proved to be fast, adaptable, and accurate.

Future work.

- **Digits in Passwords:** Future research should delve deeper into the analysis of digits and numeric-only passwords, focusing on how they correlate with alphabetic components within passwords.
- **Crowd sourced labelling:** Leveraging the collective effort of volunteers can enhance the accuracy and volume of password labelling. By enabling a broad user base to contribute labels, we can capture a wider range of languages, cultural nuances, and password constructions that individual labellers might overlook.
- **Incorporating Search Engine Context:** Integrating search engine results into the fine-tuning stage of language models can provide additional context for understanding password choices. This method could help in identifying common phrases, topical keywords, or culturally relevant terms that influence password creation, thereby enhancing the model’s predictive accuracy.
- **Improved Labelling Tools:** Developing better software and user interfaces for the password labelling process can significantly reduce the time and effort required.

References

- [1] K. Lee, S. Sjöberg, and A. Narayanan, “Password policies of most top websites fail to follow best practices,”

- in *Proceedings of the Eighteenth USENIX Conference on Usable Privacy and Security*, ser. SOUPS'22. USA: USENIX Association, Aug. 2022, pp. 561–580.
- [2] J. Bonneau, C. Herley, P. C. v. Oorschot, and F. Stajano, “The quest to replace passwords: A framework for comparative evaluation of web authentication schemes,” in *2012 IEEE Symposium on Security and Privacy*, May 2012, p. 553–567.
 - [3] B. Johnson and S. Francisco, “Hotmail password breach blamed on phishing attack,” *The Guardian*, Oct. 2009. [Online]. Available: <https://www.theguardian.com/technology/2009/oct/06/hotmail-phishing>
 - [4] J. Bonneau, “The science of guessing: Analyzing an anonymized corpus of 70 million passwords,” in *2012 IEEE Symposium on Security and Privacy*, May 2012, p. 538–552.
 - [5] A. Narayanan and V. Shmatikov, “Fast dictionary attacks on passwords using time-space tradeoff,” in *Proceedings of the 12th ACM conference on Computer and communications security*, ser. CCS '05. New York, NY, USA: Association for Computing Machinery, Nov. 2005, p. 364–372, https://www.cs.utexas.edu/~shmat/shmat_ccs05pwd.pdf. [Online]. Available: <https://dl.acm.org/doi/10.1145/1102120.1102168>
 - [6] M. Xu, C. Wang, J. Yu, J. Zhang, K. Zhang, and W. Han, “Chunk-level password guessing: Towards modeling refined password composition representations,” in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '21. New York, NY, USA: Association for Computing Machinery, Nov. 2021, p. 5–20. [Online]. Available: <https://dl.acm.org/doi/10.1145/3460120.3484743>
 - [7] “Fiverr,” <https://www.fiverr.com>, 2024.
 - [8] “National cyber security centre,” <https://www.ncsc.gov.uk>, 2024. [Online]. Available: <https://www.ncsc.gov.uk>
 - [9] “National institute of standards and technology,” <https://www.nist.gov>, Feb. 2024, last Modified: 2024-02-07T09:49-05:00. [Online]. Available: <https://www.nist.gov>
 - [10] I. McCormack, “Three random words or #thinkrandom.” [Online]. Available: <https://www.ncsc.gov.uk/blog-post/three-random-words-or-thinkrandom-0>
 - [11] NCSC, “Password policy: updating your approach,” <https://www.ncsc.gov.uk/collection/passwords/updates/your-approach>, 2018. [Online]. Available: <https://www.ncsc.gov.uk/collection/passwords/updates/your-approach>
 - [12] R. Kate, “The logic behind three random words,” Aug. 2021, <https://www.ncsc.gov.uk/blog-post/the-logic-behind-three-random-words>. [Online]. Available: <https://www.ncsc.gov.uk/blog-post/the-logic-behind-three-random-words>
 - [13] P. Grassi, M. Garcia, and J. Fenton, “Digital identity guidelines,” Mar. 2020. [Online]. Available: <https://csrc.nist.gov/pubs/sp/800/63/3/upd2/final>
 - [14] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, “QLoRA: Efficient Finetuning of Quantized LLMs,” May 2023. [Online]. Available: <https://arxiv.org/abs/2305.14314v1>
 - [15] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, “Zipf’s law in passwords,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, p. 2776–2791, Nov. 2017.
 - [16] H. Kobayashi, B. L. Mark, and W. Turin, *Probability, Random Processes, and Statistical Analysis: Applications to Communications, Signal Processing, Queueing Theory and Mathematical Finance*. Cambridge: Cambridge University Press, 2011. [Online]. Available: <https://www.cambridge.org/core/books/probability-random-processes-and-statistical-analysis/1909C657E4758038B54C4235B3AD0FDF>
 - [17] M. Weir, S. Aggarwal, B. d. Medeiros, and B. Glodek, “Password Cracking Using Probabilistic Context-Free Grammars,” in *2009 30th IEEE Symposium on Security and Privacy*, May 2009, pp. 391–405, ISSN: 2375-1207.
 - [18] —, “Password cracking using probabilistic context-free grammars,” in *2009 30th IEEE Symposium on Security and Privacy*, May 2009, p. 391–405, <https://ieeexplore.ieee.org/document/5207658>.
 - [19] G. Orwell and E. Fromm, *1984*, centennial ed.; [nachdr. der ausg.] 1. signet classics printing, july 1950, 125. [nachdr.], new american library, a division of penguin group (usa), new york, ny, 1977 ed., ser. Signet classics. St. Louis, Mo.: Turtleback Books, 2000.
 - [20] J. Steube, “Hashcat - advanced password recovery,” Oct. 2023. [Online]. Available: <https://hashcat.net>
 - [21] “KoreLogic - Home.” [Online]. Available: <https://korelogic.com>
 - [22] T. D. Tangent, “defcon.org.” [Online]. Available: <https://defcon.org/>
 - [23] J. Steube, “Hashcat - advanced password recovery,” Jun. 2023. [Online]. Available: <https://hashcat.net>

- [24] “hashcat/team-hashcat,” hashcat, Tech. Rep., Dec. 2023, original-date: 2021-08-13T18:07:16Z. [Online]. Available: <https://github.com/hashcat/team-hashcat>
- [25] D. L. Wheeler, “zxcvbn: Low-Budget password strength estimation,” in *25th USENIX Security Symposium (USENIX Security 16)*. Austin, TX: USENIX Association, Aug. 2016, pp. 157–173. [Online]. Available: <https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/wheeler>
- [26] E. Gerlitz, M. Häring, and M. Smith, “Please do not use !?_ or your License Plate Number: Analyzing Password Policies in German Companies,” 2021, pp. 17–36. [Online]. Available: <https://www.usenix.org/conference/soups2021/presentation/gerlitz>
- [27] S. Preibusch and J. Bonneau, “The Password Game: Negative Externalities from Weak Password Practices,” in *Decision and Game Theory for Security*, ser. Lecture Notes in Computer Science, T. Alpcan, L. Buttyán, and J. S. Baras, Eds. Berlin, Heidelberg: Springer, 2010, pp. 192–207.
- [28] D. Wang and P. Wang, “The emperor’s new password creation policies:,” in *Computer Security – ESORICS 2015*, ser. Lecture Notes in Computer Science, G. Pernul, P. Y. A. Ryan, and E. Weippl, Eds. Cham: Springer International Publishing, 2015, p. 456–477, <https://eprint.iacr.org/2015/825.pdf>.
- [29] D. Wang, X. Shan, Q. Dong, Y. Shen, and C. Jia, “No single silver bullet: Measuring the accuracy of password strength meters,” 2023, p. 947–964. [Online]. Available: <https://www.usenix.org/conference/usenixsecurity23/presentation/wang-ding-silver-bullet>
- [30] “Github signup,” <https://github.com/signup>, 2024, accessed: Jan 26, 2024.
- [31] “Gov.uk hmrc registration,” <https://www.gov.uk/log-in-register-hmrc-online-services/register>, 2024, accessed: Jan 27, 2024.
- [32] “Yahoo account creation,” <https://login.yahoo.com/account/create?.intl=uk&.lang=en-GB>, 2024, accessed: Jan 28, 2024.
- [33] “Proton mail signup,” <https://account.proton.me/mail/signup?plan=free>, 2024, accessed: Jan 29, 2024.
- [34] “Reddit registration,” <https://www.reddit.com/register>, 2024, accessed: Jan 30, 2024.
- [35] “X signup,” <https://twitter.com/i/flow/signup?lang=en>, 2024, accessed: Jan 31, 2024.
- [36] “Microsoft account signup,” <https://signup.live.com/signup>, 2024, accessed: Jan 26, 2024.
- [37] “Spotify signup,” <https://www.spotify.com/uk/signup>, 2024, accessed: Jan 27, 2024.
- [38] “Roblox,” <https://www.roblox.com/>, 2024, accessed: Jan 28, 2024.
- [39] “ebay signup,” <https://signup.ebay.co.uk/pa/crte>, 2024, accessed: Jan 29, 2024.
- [40] “Google account creation,” <https://accounts.google.com>, 2024, accessed: Jan 30, 2024.
- [41] “Ealing council signin,” <https://www.ealing.gov.uk/signin>, 2024, accessed: Jan 31, 2024.
- [42] “Royal borough of kensington and chelsea account creation,” <https://www.rbkc.gov.uk/myrbkc-account/create-myrbkc-account>, 2024, accessed: Jan 26, 2024.
- [43] “Nhs login,” <https://www.nhsapp.service.nhs.uk/login>, 2024, accessed: Jan 27, 2024.
- [44] Facebook, “Facebook,” <https://www.facebook.com>, 2024, accessed: Feb 4, 2024.
- [45] “Instagram signup,” <https://www.instagram.com/accounts/emailsignup>, 2024, accessed: Jan 28, 2024.
- [46] “Linkedin signup,” <https://www.linkedin.com/signup>, 2024, accessed: Jan 29, 2024.
- [47] “Netflix signup,” <https://www.netflix.com/gb/signup>, 2024, accessed: Jan 30, 2024.
- [48] “Amazon,” <https://www.amazon.co.uk>, 2024, accessed: Jan 31, 2024.
- [49] OpenAI, “Gpt-3.5 turbo fine-tuning and api updates,” <https://openai.com/blog/gpt-3-5-turbo-fine-tuning-and-api-updates>. [Online]. Available: <https://openai.com/blog/gpt-3-5-turbo-fine-tuning-and-api-updates>
- [50] H. Touvron, L. Martin, K. R. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. M. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. M. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom, “**Llama 2: Open Foundation and Fine-Tuned Chat Models**,” *ArXiv*,

- Jul. 2023, <https://arxiv.org/pdf/2307.09288.pdf>. [Online]. Available: <https://www.semanticscholar.org/paper/Llama-2%3A-Open-Foundation-and-Fine-Tuned-Chat-Models-Touvron-Martin/104b0bb1da562d53cbda87aec79ef6a2827d191a>
- [51] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed, “**Mistral 7B**,” no. arXiv:2310.06825, Oct. 2023. [Online]. Available: <http://arxiv.org/abs/2310.06825>
- [52] N. Pangakis, S. Wolken, and N. Fasching, “Automated annotation with generative ai requires validation,” no. arXiv:2306.00176, May 2023, arXiv:2306.00176 [cs]. [Online]. Available: <http://arxiv.org/abs/2306.00176>
- [53] Y. Liu, “The importance of human-labeled data in the era of llms,” no. arXiv:2306.14910, Jun. 2023, arXiv:2306.14910 [cs]. [Online]. Available: <http://arxiv.org/abs/2306.14910>
- [54] A. G. Møller, J. A. Dalsgaard, A. Pera, and L. M. Aiello, “The parrot dilemma: Human-labeled vs. llm-augmented data in classification tasks,” no. arXiv:2304.13861, Feb. 2024, <http://arxiv.org/abs/2304.13861>. [Online]. Available: <http://arxiv.org/abs/2304.13861>
- [55] “Llms can label data as well as humans, but 100x faster,” url={<https://refuel.ai/blog-posts/llm-labeling-technical-report>}, 2024. [Online]. Available: <https://refuel.ai/blog-posts/llm-labeling-technical-report>
- [56] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, “Lora: Low-rank adaptation of large language models,” Jun. 2021, <https://arxiv.org/pdf/2106.09685.pdf>. [Online]. Available: <https://arxiv.org/abs/2106.09685v2>
- [57] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, “Qlora: Efficient finetuning of quantized llms,” May 2023, <https://arxiv.org/pdf/2305.14314.pdf>. [Online]. Available: <https://arxiv.org/abs/2305.14314v1>
- [58] S. Mukherjee, A. Mitra, G. Jawahar, S. Agarwal, H. Palangi, and A. Awadallah, “Orca: Progressive learning from complex explanation traces of gpt-4,” 2023.
- [59] A. N. Lee, C. J. Hunter, and N. Ruiz, “Platypus: Quick, cheap, and powerful refinement of llms,” 2023.
- [60] S. Es, “Orca-best: A filtered version of orca gpt4 dataset,” <https://huggingface.co/datasets/shahules786/orca-best>, 2023.
- [61] L. Berglund, M. Tong, M. Kaufmann, M. Balesni, A. C. Stickland, T. Korbak, and O. Evans, “The reversal curse: Llms trained on “a is b” fail to learn “b is a”,” no. arXiv:2309.12288, Sep. 2023, arXiv:2309.12288 [cs]. [Online]. Available: <http://arxiv.org/abs/2309.12288>
- [62] L. d. Castro, H. Lang, S. Liu, and C. Mata, “Modeling password guessing with neural networks,” 2017. [Online]. Available: <https://www.semanticscholar.org/paper/Modeling-Password-Guessing-with-Neural-Networks-Castro-Lang/8ef01b61ec9428556ba78465f4098baf7734f613>
- [63] B. Hitaj, P. Gasti, G. Ateniese, and F. Perez-Cruz, “Passgan: A deep learning approach for password guessing,” in *Applied Cryptography and Network Security*, vol. 11464. Springer, 2019, p. 217–237, accepted: 2020-02-13T13:21:35Z. [Online]. Available: <https://www.research-collection.ethz.ch/handle/20.500.11850/386747?locale-attribute=de>
- [64] W. Melicher, B. Ur, S. M. Segreti, S. Komanduri, L. Bauer, N. Christin, and L. F. Cranor, “Fast, lean, and accurate: Modeling password guessability using neural networks,” 2016, p. 175–191. [Online]. Available: <https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/melicher>
- [65] M. Xu, J. Yu, X. Zhang, C. Wang, S. Zhang, H. Wu, and W. Han, “Improving real-world password guessing attacks via bi-directional transformers,” 2023, p. 1001–1018. [Online]. Available: <https://www.usenix.org/conference/usenixsecurity23/presentation/xu-ming>
- [66] J. Ma, W. Yang, M. Luo, and N. Li, “A study of probabilistic password models,” in *2014 IEEE Symposium on Security and Privacy*, May 2014, p. 689–704, <https://ieeexplore.ieee.org/document/6956595>. [Online]. Available: <https://ieeexplore.ieee.org/document/6956595>
- [67] Z. Li, W. Han, and W. Xu, “A large-scale empirical analysis of chinese web passwords,” Aug. 2014. [Online]. Available: <https://www.semanticscholar.org/paper/A-Large-Scale-Empirical-Analysis-of-Chinese-Web-Li-Han/e165c258694cf774690fc06564c9bf6b0fb7fdb8>
- [68] A. Das, J. Bonneau, M. Caesar, N. Borisov, and X. Wang, “The tangled web of password reuse,” in *Proceedings 2014 Network and Distributed System Security Symposium*. San Diego, CA: Internet Society, 2014. [Online]. Available: <https://www.ndss-symposium.org/ndss2014/programme/tangled-web-password-reuse/>
- [69] B. Ur, S. M. Segreti, L. Bauer, N. Christin, L. F. Cranor, S. Komanduri, D. Kurilova, M. L. Mazurek, W. Melicher,

and R. Shay, “Measuring real-world accuracies and biases in modeling password guessability,” 2015, p. 463–481, <https://www.usenix.org/system/files/conference/usenixsecurity15/sec15-paper-ur.pdf>. [Online]. Available: <https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/ur>

- [70] Y. Li, H. Wang, and K. Sun, “A study of personal information in human-chosen passwords and its security implications,” in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, Apr. 2016, p. 1–9, <https://ieeexplore.ieee.org/document/7524583>. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/7524583>
- [71] S. Pearman, J. Thomas, P. E. Naeini, H. Habib, L. Bauer, N. Christin, L. F. Cranor, S. Egelman, and A. Forget, “Let’s go in for a closer look: Observing passwords in their natural habitat,” *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, p. 295–310, Oct. 2017, <https://dl.acm.org/doi/10.1145/3133956.3133973>. [Online]. Available: <https://dl.acm.org/doi/10.1145/3133956.3133973>
- [72] D. Pasquini, M. Cianfriglia, G. Ateniese, and M. Bernaschi, “Reducing bias in modeling real-world password strength via deep learning and dynamic dictionaries,” 2021, p. 821–838, <https://www.usenix.org/conference/usenixsecurity21/presentation/pasquini>. [Online]. Available: <https://www.usenix.org/conference/usenixsecurity21/presentation/pasquini>
- [73] J. A. Cazier and B. D. Medlin, “Password security: An empirical investigation into e-commerce passwords and their crack times,” *Information Systems Security*, vol. 15, no. 6, p. 45–55, Dec. 2006.
- [74] C. Bonk, Z. Parish, J. Thorpe, and A. Salehi-Abari, “Long passphrases: Potentials and limits,” no. arXiv:2110.08971, Oct. 2021, <https://arxiv.org/pdf/2110.08971.pdf>. [Online]. Available: <http://arxiv.org/abs/2110.08971>
- [75] Y. Li, H. Wang, and K. Sun, “A study of personal information in human-chosen passwords and its security implications,” in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, Apr. 2016, p. 1–9. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/7524583>
- [76] B. L. Riddle, M. S. Miron, and J. A. Semo, “Passwords in use in a university timesharing environment,” *Computers & Security*, vol. 8, no. 7, p. 569–579, Nov. 1989.
- [77] D. Malone and K. Maher, “Investigating the distribution of password choices,” in *Proceedings of the 21st international conference on World Wide Web*, ser.

WWW ’12. New York, NY, USA: Association for Computing Machinery, Apr. 2012, p. 301–310. [Online]. Available: <https://doi.org/10.1145/2187836.2187878>

- [78] J. Rando, F. Perez-Cruz, and B. Hitaj, “Passgpt: Password modeling and (guided) generation with large language models,” no. arXiv:2306.01545, Jun. 2023, <http://arxiv.org/abs/2306.01545>. [Online]. Available: <http://arxiv.org/abs/2306.01545>

A Supporting Material

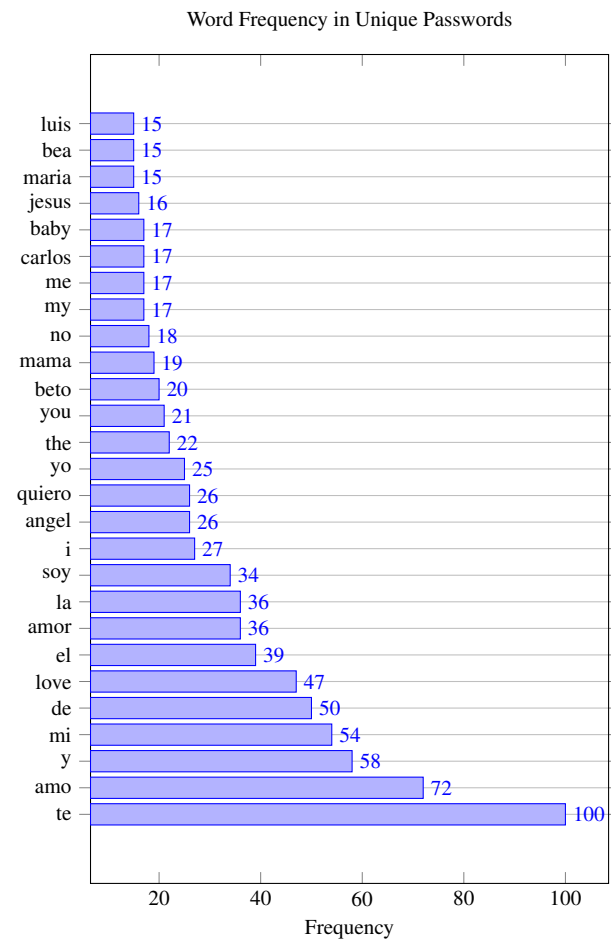


Figure 8: Chart showing the frequency of individual words regardless of their tag. These words could be tagged as word, name, location or be part of a phrase. These words are extracted from unique passwords in the hotmail2009 dataset.