LLM-Generated Passphrases That Are Secure and Easy to Remember

Anonymous ACL submission

Abstract

Automatically generated passwords and passphrases are a cornerstone of IT security. Yet, these passphrases are often hard to remember and see only limited adoption. In this work, we use large language models to generate passphrases with rigorous security guarantees via the computation of the entropy of the output as a metric of the security of the passphrase. We then present a range of practical methods to generate language model outputs with sufficient entropy: raising entropy through in-context examples and generation through a new top-q truncation method. We further verify the influence of prompt construction in steering the output topic and grammatical structure. Finally, we conduct user studies to determine the adoption rates for these LLM-generated passphrases in practice.

1 Introduction

Automated passphrase generation is an effective approach to ensure that users have secure passphrases and that passwords are not reused across applications. While it offers undeniable security benefits, randomly generated passphrases are commonly more challenging to remember due to their lack of semantic meaning (Meng et al., 2021), leading to disinterest from users and other security-threatening behaviors like leaving passphrases exposed on post-it notes.

In this paper, we explore the use of large language models (LLMs) as a tool for generating randomized passphrases that maintain semantic coherence. We demonstrate that autoregressive LLMs possess several advantageous properties that can be leveraged for passphrase creation. Notably, the LLM functions as a probabilistic model of language and can be used to compute the entropy score for each passphrase. This score quantifies the number of bits of entropy present in a passphrase, providing a rigorous and interpretable metric for assessing password security. Furthermore, by incorporating user inputs and preferences into prompts, LLMs can generate passphrases tailored to specific topics (Meng et al., 2021).

A surprising, but crucial, fact of this construction is that model-evaluated passphrase entropy is indeed a *rigorous* measure of security. As such, these generated passphrases do not merely "appear random", but their randomness is exactly quantifiable.

Despite the capabilities of today's open-source language models, generating secure passphrases remains a complex task. Care must be taken to ensure that the generated phrases guarantee sufficient entropy to be comparable in security to other randomized passphrase generation schemes. In this paper, we address several key issues related to passphrase generation. We begin with a discussion of how to compute entropy for a passphrase. We then investigate how to optimize passphrase generation for each base LLM by utilizing various prompt engineering strategies to select effective prompt format and content and by selecting generation parameters to maximize the number of samples meeting entropy requirements.

In summary, our goal is to generate passphrases that satisfy the following two important criteria:

- *Usability:* In practice, people often ignore guidelines that ensure the security of passwords and passphrases in favor of selecting weaker and easier to remember passwords, (Wang et al., 2017) and so usability is critical. Passphrases should be convenient and pleasant to use. For example, the password "uEBJd6n35", created by an online password generator (Avast, 2024), is hard to remember, challenging to pronounce, and potentially avoided by users. It is then advantageous to use LLMs to generate passphrases as their text

output is aligned to natural language, making the output especially amenable for human users to remember and to repeatedly input.

- *Security:* The randomness of the LLM sampling process can be leveraged to produce output that is hard-to-guess. We introduce a metric to quantify this hard-to-guess-ness in the definition of entropy in Section 3.

2 Related Works

We use LLMs to generate secure passphrases, while recent work has used LLMs to evaluate and attack passwords. (Jin et al., 2024) generates passwords from a neural network in decreasing order of probability to build a password cracking system. (Wang et al., 2023) uses a generative model to characterize users password modification behaviors in the context of password tweaking attacks. (Tan et al., 2020a) uses a neural network trained on leaked password data to consider minimum-strength requirements.

The challenges of meeting security guidelines in the real-world setting is well-documented. (Shay et al., 2016) shows user difficulty in remembering passwords that conform to password composition policies. Even website administrators often fail to follow good practice in password guidelines (Lee et al., 2022).

Together this shows the need for and the advantage that a well-crafted generation method can offer. (Wu et al., 2022) shows computer-generated passphrases are more diverse than human-generated passphrases in a user-survey conducted at a major university which uses passphrases for its login.

3 Entropy Definition

Shannon entropy (Shannon, 1948) has been used in password security measure through applying a uniform probability distribution over letter and digits (Komanduri et al., 2011). In contrast, we use the distribution of the language model generation.

We define entropy of a sequence of tokens from a language model to be $\mathcal{H} = -\log_2(P)$, where P is the probability of the sequence of tokens. More formally, for sequence $w_1, w_2, \ldots, w_n$, we have

$$P(w_1, w_2, \ldots, w_n) = \prod_{i=1}^{n} P(w_i \mid w_1, \ldots, w_{i-1}) \tag{1}$$

where the probability values results from the softmax function applied to the logits of the language model. Here, $P(w_1)$ denotes the probability of the first token, $P(w_2 \mid w_1)$ is the probability of the second token conditioned on the first token, $P(w_3 \mid w_1, w_2)$ is the probability of the third token conditioned on the first two tokens, etc. Note that in practice we often further condition text on a prompt, but we have omitted this in our notation for brevity.

3.1 Entropy measures security and diversity

The probability $P(w_1, w_2, \ldots, w_n)$ defined above measures how likely our language model is to produce a particular sequence of tokens. The entropy, $\mathcal{H} = -\log_2(P)$, tells us how many bits of randomness are present in a text sequence.

If a randomly generated password contains b bits of randomness, then an optimal strategy for guessing the password requires 2^b attempts in expectation. Likewise, brute-forcing an LLM generated password requires $2^{\mathcal{H}}$ guesses in expectation, even in the white box scenario where the LLM is available to the adversary. Using entropy, we can directly compare LLM generated passphrases to randomly sampled passphrases of equivalent security.

Note also that computing entropy protects us from a bad LLM that tends to reproduce the same phrases repeatedly; these phrases will have low entropy, and can be discarded after generation to ensure secure passwords. If the user is allowed to provide inputs to influence the password generation algorithm, an entropy check can prevent insecure passwords from being produced, even if the user provides a pathological input that collapses the model output to a single mode.

3.2 Relationship between entropy and perplexity

Perplexity is a metric of model performance, commonly used to compare the same output across two models. The definition of Perplexity for a sequence of tokens, $w_1, w_2, \ldots, w_n$ with probability $P(w_1, w_2, \ldots, w_n)$, is

$$\begin{aligned} \text{Perplexity} &= (P(w_1, w_2, \ldots, w_n))^{-\frac{1}{n}} \\ &= 2^{\frac{\mathcal{H}}{n}} \end{aligned} \tag{2}$$

Both Perplexity and our definition of entropy capture how likely a model is to output a text sequence, but Perplexity is normalized by n, the

number of tokens making up the text. This means that if two models output a piece of text with the same probability but tokenize the text with different number of tokens, then they would have the same entropy but different Perplexity for that text. In contrast, entropy as a metric stays the same.

4 Passphrase Generation and Criteria

4.1 Passphrase criteria

We use three criteria to control the distribution of generated passphrases. Each passphrase must (i) contain a minimum allowable entropy to ensure security. It should contain (ii) fewer than maximum allowable number of words to make it practical. Finally (iii) it should only contain correctly spelled English words. This last criteria is important as most LLMs can output sequences of tokens that are difficult for humans to interpret, or contain non-standard typography (e.g., emojis). While not a formal criteria, another underlying goal is to create passphrases that tell a story or parable to aid memorability.

In our experiments we consider passwords with a minimum of 47 bits of entropy (and some of our generations reach as high as 80 bits). This level of security is recommended by (Tan et al., 2020b) and (Xu et al., 2021), and it requires 10^{14} trials (on average) to attack by brute force. We also require that the passphrase be eight or fewer words so that they do not become overly cumbersome. We disallow punctuation other than apostrophe and dash and disallowing digits and non-ascii characters through, for the most part (with details in Appendix A.9), by setting the probability of tokens that contain such characters to be zero via adjusting the value of the logits. This step is not applied during the experiments in Section 5 as we want to examine the output without additional adjustments of logits.

4.2 Text generation settings

We generate texts using common open-source chat/instruct models models, with a list of full model names in Appendix A.1.

We use multinominal sampling while setting the parameters: temperature and top-p. We also introduce another sampling parameter called top-q truncation to be discussed. Higher temperature and higher top-p parameters tend to result in higher entropy values for the output. We select the parameter values based on the desired entropy target range and set temperature to be 1.0, 1.2 and 1.4 and top-p to be 0.95, 0.99, and 1.0, unless specifically otherwise noted, and sampled ≥ 128 for each parameter combination. Note that we compute entropy scores using the probability distribution for tokens *after* any temperature scaling, top-p sampling, or other sampling schemes are applied.

4.3 Entropy correction for rejection sampling

Below, we discuss the generation of passphrases through rejection sampling, in which each LLM generated passphrase is tested for the above criteria, and then it is tossed out and regenerated if it fails. This process increases the likelihood of generating certain passwords. For example, if we reject and regenerate half of the passwords for containing too little entropy, then this doubles the probability of outputting any one of high-entropy passwords.

If our sampling process requires n attempts to produce a passphrase, then we subtract $\log_2(n)$ bits from the entropy to compensate.

5 Experiments

We perform experiments on the effects of prompt format and content and on methods to increase the entropy of the output sample.

5.1 Direct-question prompt versus template form

Prompts such as “Give me a story in six words" or other direct prompts are ineffective at generating good passphrases as the prompt is too open-ended and there is not enough information or guidance of the desired output. Table 1 shows that a low percentage of the output meet the entropy requirement (entropy $\geq$ 47). See Tables 11and 12 in the Appendix for output percentage that meet the requirements on number of words (eight or fewer words) and English language (correctly-spelt words) criteria.

Note that many of the model outputs are prefaced with low-entropy words like “Sure" “Certainly", which contribute to the prompt length without adding much security. See Appendix A.3 for a study of this and also for results at other temperature and top-p settings. Overall, direct-question prompts are ineffective at generating passphrases.

Language models are few-shot learners, and providing in-context examples in the prompt can be crucial (Brown et al., 2020) to raising entropy. Similarly, in-context examples help to guide the model towards the format of the desired response (Min

model	story	passp	sum	TLDR
Llama-2-7B	0.0	0.4	0.0	0.1
Llama-2-13B	0.1	0.3	0.1	0.2
Llama-2-70B	0.1	0.3	0.2	0.2
Llama-3-8B	0.1	0.7	0.2	0.2
Llama-3-70B	0.0	0.1	0.0	0.0
Mistral-7B	0.9	0.9	0.6	0.7
Mixtral-8x7B	0.4	0.4	0.3	0.2
gemma-7B	0.3	0.1	0.0	0.0

Table 1: For direction-question prompts, proportion of outputs that have entropy ≥ 47. "story" column corresponds to prompt of "Give me a story in six words"; "passp" to "Write a passphrase in six words"; "sum" to "Write a summary of a story in six words"; "TLDR" to "Write a tldr of a story in six words" (sampled with temperature of 1.4 and top-p of 0.95)

.

et al., 2022). We show the importance of providing examples and a template for output in the prompt construction in the below comparison of the "direct-question" prompt, whose failings we note above, versus the template form which is show in Figure 1.

Specifically we fill the template (Figure 1) with six example user and assistant interactions from Table 2 and use this as a prompt, called the base-prompt, to generate passphrases. See Figure 2 for sample responses to the base-prompt. See Table 22 for results. These associated output texts meet many more of the three passphrase criteria than the direct-question prompts.

Increasing the temperature or the top-p parameters increases the output sample average entropy, but outputs meeting the number of words criteria or the only English words may decrease. See Appendix A.4 for additional results using the base prompt at different generation settings.

5.2 Higher entropy output through higher entropy input

How much modern LLMs are capable of learning from in-context examples is a current research question. (Min et al., 2022) argues the general distributional properties may be learned rather than specific correct labeling. Here we show that the distribution of the entropy of the input examples is learned and mimicked in the output. We create a prompt using the template and fill it with examples with low entropy, and we create another prompt

Figure 1: Prompt following template showing two examples. This prompt follows the template format and all prompts in this paper, except for the direct-question prompts, follow this format but may have a different number of examples.

User	Assistant
Give the summary of a story in six or fewer words.	Multiple erasers banish innovation sparks
Give the summary of a story in six or fewer words.	Hassled celebrity resists spaceship visit
Give the summary of a story in six or fewer words.	Fabulous doctor maintained patient dreams
Give the summary of a story in six or fewer words.	Television shoot overflows fruit boat capacity
Give the summary of a story in six or fewer words.	Pavlov patronizes papaya pyramid king
Give the summary of a story in six or fewer words.	Courteous yeti plays quiet prodigy
Give the summary of a story in six or fewer words.	

Table 2: These are the examples used to fill the prompt template to create the base prompt.

model	entropy $\geq$ 47.0	num words ≤ 8	Eng	All criteria
Llama-2-7B	0.8	1.0	0.8	0.6
Llama-2-13B	0.9	1.0	0.8	0.7
Llama-2-70B	0.7	0.9	0.6	0.3
Llama-3-8B	0.9	1.0	0.6	0.6
Llama-3-70B	0.3	1.0	0.8	0.1
Mistral-7B	1.0	0.7	0.6	0.6
Mixtral-8x7B	1.0	0.9	0.7	0.7
gemma-7B	0.9	0.7	0.5	0.3

Table 3: Proportion of output that meet criteria of entropy $\geq$ 47.0, num_words $\leq$ 8 and consisting of all English words, separately, and all criteria concurrently using the base prompt, temperature of 1.4 and top-p of 0.95.

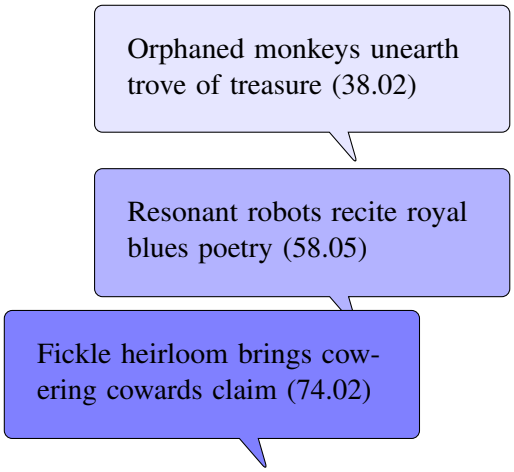

Figure 2: Mistral speaks! (Mistral responses at different entropy values with entropy in parenthesis at temp 1.2 and top-p 0.95 for base-prompt.)

using the template and fill it with examples with high entropy.

See Table 8 for average entropy comparisons between outputs from base prompts versus the low entropy prompts and the high entropy prompts. Higher entropy input examples tend to promote high entropy output examples. The strength of this tendency varies with temperature and top-p settings. See Appendix A.6.

5.3 Higher entropy output through top-q truncation sampling

Entropy can be influenced by setting the temperature and top-p parameters. The top-p parameter zeroes out the probability of the tokens outside of the top p of the distribution (Holtzman et al., 2019). Specifically, given a probability distribution $P(x \mid x_{1:i-1})$ over the vocabulary V, the top-p vocabulary V_p with $V_p \subseteq V$ is the smallest set such that:

$$\sum_{x \in V_p} P(x \mid x_{1:i-1}) \geq p \tag{3}$$

where p is the top-p parameter, with $0 < p \leq 1$. The tokens not in V_p are assigned a probability of 0 and the probability of the tokens in V_p are renormalized to sum to 1. Then the nucleus sampling method selects a token from V_p per the normalized distribution. (Note: in implementation code, the $\geq p$ of Equation 3 is often replaced with $> p$ for floating-point arithmetic computation.)

We introduce a method to increase entropy through an analogous method, the top-q truncation method, which zeroes out the top q of the distribution. For example, if one token occupies 0.88 probability mass with all others the remaining 0.12, then a top-q truncation of 0.05 would decrease the first token mass to 0.83 and leave all others untouched, then the distribution is renormalized and sampled from.

This eliminates the most common outputs, forcing the model to choose creative words rather than defaulting to the most obvious (and low entropy) choice. By zeroing out the top-q portion of the distribution, this raises the sample output entropy distribution, and causes far fewer samples to be rejected. More generally this method can be used to create more diverse output without sacrificing coherence.

See Table 4 for a comparison between no q-value application (q-value of 0) versus various top-q truncation at various q-values. Average entropy of the output increases as q-value increases.

5.4 Steering output topic through input content

As discussed, to be tenable to users, there must be options provided to personalize user passphrases. The task of steering the topic or other characteristics of the LLM output has been approached in various ways, such as described in (Dathathri et al., 2019) and (Sanchez et al., 2023). In our prompt template, a significant degree of steering of the output topic is possible by providing examples of the topic in the prompt. To demonstrate, we create a prompt containing examples with cat-related words. The output has more “cat", “cats" and cats-related words than the base prompt, which is not related to

model	$q =$ 0.0	$q =$ 0.05	$q =$ 0.20	$q =$ 0.35
Llama-2-7B	30.4	31.3	34.9	38.8
Llama-2-13B	35.0	36.3	39.2	40.0
Llama-2-70B	21.5	22.8	24.8	27.6
Llama-3-8B	32.7	35.2	37.6	41.6
Llama-3-70B	16.5	17.2	19.6	21.1
Mistral-7B	43.4	44.9	48.6	49.5
Mixtral-8x7B	49.5	53.7	53.0	52.1
gemma-7B	34.8	36.5	40.4	43.6

Table 4: Average entropy of output at various q values. (Sample size of 128 using base prompt with temperature of 1.0 and top-p of 0.95)

model	base prompt cat output	base prompt purr output	cats prompt cat output	cats prompt purr output
Llama-2-7B	0.06	0.02	0.67	0.17
Llama-2-13B	0.00	0.00	0.64	0.29
Llama-2-70B	0.01	0.01	0.81	0.15
Llama-3-8B	0.01	0.00	0.75	0.10
Llama-3-70B	0.00	0.01	0.96	0.01
Mistral-7B	0.00	0.00	0.87	0.15
Mixtral-8x7B	0.01	0.00	0.86	0.07
gemma-7B	0.04	0.01	0.52	0.22

Table 5: Proportion of output that contains "cat" or "cats" and proportion of output that contains "purr" or other cat-related words ("feline" or "kitten"), compared between base prompt and cats-prompt.

cats. See Table 5. See Appendix A.5 for examples of steering towards other topics.

5.5 Steering output part-of-speech construction

Language models have been used on tasks that require knowledge of grammar (Lakretz et al., 2022) (Lampinen, 2024) (Wang et al., 2024), and so it is not surprising that they can mimic the grammatical structure of in-context examples. We demonstrate this through two example prompts. The first prompt contains examples that start with an adjective and a noun to be followed by other words (the adjective-noun prompt) and the second contains examples that start with a noun and a verb to be followed by other words (the noun-verb prompt).

For the prompts, we look at the models output and count the number of instances which start

model	base prompt	adj-noun prompt	noun-verb prompt
Llama-2-7B	0.37	0.45	0.24
Llama-2-13B	0.37	0.50	0.13
Llama-2-70B	0.01	0.01	0.01
Llama-3-8B	0.37	0.63	0.20
Llama-3-70B	0.54	0.82	0.09
Mistral-7B	0.32	0.33	0.15
Mixtral-8x7B	0.32	0.44	0.24
gemma-7B	0.26	0.27	0.30

Table 6: Proportion of output that start with adjective-noun.

model	base prompt	adj-noun prompt	noun-verb prompt
Llama-2-7B	0.08	0.07	0.22
Llama-2-13B	0.05	0.03	0.12
Llama-2-70B	0.01	0.00	0.02
Llama-3-8B	0.12	0.02	0.23
Llama-3-70B	0.02	0.01	0.29
Mistral-7B	0.09	0.05	0.29
Mixtral-8x7B	0.07	0.05	0.23
gemma-7B	0.05	0.09	0.11

Table 7: Proportion of output that start with noun-verb.

with leading adjective-noun structure or with leading noun-verb structure. We compare the base prompt, the adjective-noun prompt and the noun-verb prompt. See Table 6 for increased occurrence of adjective-noun output for adjective-noun prompt relative to the other two prompts and see Table 7 for increased occurrence of nounverb output for nounverb prompt relative to the other two prompts.

Note that the part-of-speech of each output word is determined using Spacy's English pipeline (Honnibal et al., 2020).

5.6 User study in LLM passphrases memorability

We perform a series of *user studies* to evaluate the human memorability of the LLM passphrases. We choose a baseline of random-word passphrases, with each composed of five words randomly chosen with replacement from a list of 800 easy and common English words (from a subset of the word

list available at EF, 2024). The random words passphrases have an entropy of 48.2 bits. See (Wu et al., 2022) for evaluating users response to computer-generated random five-word passphrases in its study of memorability.

Section A.10 discusses the information and consent form provided to the participants. The studies are conducted at the beginning and end of a one-hour session, in the context of a meeting or a university class. All of the participants are university affiliates (students, faculty, staff). At the beginning, the participants receive the passphrase printed out on a sheet of paper and the participants are instructed to expend a reasonable effort to remember the passphrase, and then to fold the sheet of paper to hide the passphrase. At the end of the one-hour session, they are asked to reproduce the passphrase from memory and write it on the outside of the folded page.

Results from the first user study are as follows. Among a total of 16 participants, eight were assigned LLM passphrases and eight were assigned random-word passphrases. In each group, three out of eight participants correctly remembered their passphrase whereas five out of eight did not. We observed that the memorizability of both LLM and random word phrases was comparable in this study. We suspect this may be because the random-word passphrases, while lacking semantic meaning, are composed of more common words (e.g., “object money react capable willing").

6 Analysis

6.1 Model comparisons

The larger Llama models have noticeable different entropy behavior than their smaller counterparts. Llama2-70B has lower entropy at temperature of 1.0 and top-p of 1.0 than Llama2-7B and Llama2-13B. Similarly Llama3-70B has lower entropy at temperature of 1.0 and top-p of 1.0 than Llama3-8B. See Table 8. Notice this is not the case at other temperature and top-p combinations. See Table 28

To start parsing this result, we look at the Llama-2-70B logits for the first to-be-generated token and calculate the expected entropy of the first token. Technically the first token of Llama-2-70B is a space token and we evaluate here its second token. Figure reveals that Llama-2-70B has a slightly lower entropy at temperature 1.0 but has a slightly higher entropy at 1.6. Although not explaining the magnitude of the difference observed in Tables 8 and 28, it shows that changing the temperature can change the relative order of the models in output token entropy. It also illustrates the overall effect of increasing temperature on output entropy.

model	base prompt	low-ent prompt	high-ent prompt
Llama-2-7B	36.80	20.90	34.10
Llama-2-13B	40.10	19.50	37.10
Llama-2-70B	28.40	17.30	24.80
Llama-3-8B	35.80	22.40	35.00
Llama-3-70B	19.40	4.70	19.10
Mistral-7B	55.10	25.80	53.30
Mixtral-8x7B	52.50	33.40	58.60
gemma-7B	37.10	27.70	32.80

Table 8: Average entropy of output using the base prompt, prompt containing low entropy examples and prompt containing high entropy examples, with temperature of 1.0 and top-p of 1.0.

6.2 English language distribution

Common to all models is the task of generating natural English text sequence, subject to the structure of the English language, including the distribution of nouns, verbs, adjectives and others parts of speech. In English, there are more nouns than any other parts of speech (Hudson, 1994). Contrast this with random word passphrases: there the first word can be chosen from N words (N being the vocabulary size) and the second word can be chosen from N words and so on. The entropy is evenly spread out among all words in the passphrase and each word is approximately equally memorable and informative and by design are not related to one another. In contrast, for a noun-verb sequence, there are more choices of nouns than verbs, hence the entropy is more on the noun than the verb. Further the noun and verb are linked to each other grammatically and semantically. The noun-verb sequence is but one exemplar in natural language, which language model mimics.

6.3 Implementation of passphrase generation

We present a process to create passphrases that focuses on the user-experience while being measurably secure. The process allows the user to select a topic for the passphrase (e.g., cats, biology, culture), contributing to the affinity of the user to the passphrase and thereby increasing the usability of the passphrase. In addition to setting tempera-

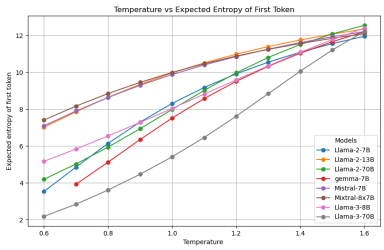

Figure 3: For the sampling of the first token, calculate the probability of all tokens. Then calculate the expected entropy for that first token at different temperatures for various models.

model	temp	top-p	q-trunc	all-criteria
Llama-2-7B	1.4	0.95	0.35	0.57
Llama-2-7B	1.4	0.95	0.00	0.48
Llama-2-13B	1.2	0.99	0.35	0.62
Llama-2-70B	1.4	1.00	0.00	0.31
Llama-3-8B	1.4	0.95	0.35	0.59
Llama-3-70B	1.4	1.00	0.35	0.27
Mistral-7B	1.2	0.95	0.35	0.65
Mixtral-8x7B	1.2	0.95	0.00	0.68
gemma-7B	1.2	1.00	0.35	0.46

Table 9: Settings for passphrase generation. For each model, use the base-prompt and set temperature, top-p, and q-truncation as in the above to create good passphrases. The last column lists the proportion of output which meets all three criteria: entropy ≥ 49, number of words ≤ 8, and composed of English words.

ture and top-p, we introduce two additional ways to ensure the security of passphrases as measured by entropy through selection of high-entropy input examples and using the new top-q truncation sampling method, described in Section 5.3

As a final step to standardize output during passphrase generation, we disallow punctuation other than apostrophe and dash and disallowing digits and non-ascii characters through, for the most part (with details in Appendix A.9), by setting the probability that tokens containing such characters to zero (via adjusting the value of the logits). This step is not applied during the experiments in Section 5 as we want to examine the output without additional adjustments of logits.

To provide specific implementation steps, for each of the models, input the base-prompt and set the generation parameters per Table 9. Not all but a large portion of the output meet the criteria of entropy ≥ 47, number of words ≤ 8 and consisting of English words. The portion that does not meet the criteria are to be discarded through automated filtering. To account for this filtering, we increase the entropy requirement. For instance, increasing the entropy by 1-bit accounts for half of the output being filtered in and increasing the entropy by 2-bits accounts for one-fourth of the output being filtered in. For all of the models at the settings in Table 9, more than one-fourth meet the criteria of maximum number of words, typographical English words, and also the increased entropy requirement of 49 bits. As a result, the filtered in sample also meet the 47-bit minimum entropy requirement.

The base-prompt, along with the settings in Table 9, offers one concrete way to generate good passphrases. It is not exhaustive and see Appendix A.8 for additional parameter settings for passphrase generation. As noted in Section 5.4, the user may also choose to create his own prompt based on his preferred topic while nothing the guide on format and content in Sections A.3 and 5.4.

To reiterate, the process consists of the following: sample the specified models at the given sequence of parameters with the base prompt or the chosen prompt of the user-specified topic following the format of the base prompt. Filter the results for the three criteria: output entropy, number of words and consisting of English words. Use an elevated entropy level to account for the filtering process.

7 Conclusion

We demonstrate how language models can be used to create user-friendly passphrases while meeting security requirements. We show the importance of the prompt choice in terms of format and content and the influence that sampling parameters have on the output.

Despite lack of consensus in the effect, effectiveness and mechanism of in-context examples, we show in the specific task of passphrase generation, in-context examples are effective at steering topic, grammatical structure, and entropy distribution.

We also introduce an additional sampling parameter, called top-q-truncation, which can guide to higher entropy values and greater diversity in outcomes.

Limitations

Our method uses instruction-tuned pre-trained models to generate text outputs and are thus dependent

on the effectiveness of the instruction-tuning to respond to the inputted prompts and on the quality of the pre-training for useful output. We present prompt templates to obtain desired outputs and a selection process to further cull output to create good passphrases. Model size and other factors affect model output quality (Kaplan et al., 2020). A limitation of our method is that it may not work on weaker models, which either respond poorly to instructions or output low-quality text sequences. In that case, we may not be able to obtain a sufficient number of output that meet all of the criteria for a good passphrase. We note that we include reasonably-sized models in our experiments, such as Llama-2-7B, Llama-3-8B, and Mistral-8B, and these are models which may be accessed by the public.

Ethics Statement

Potential Risks

Our method allows the generation of passphrases that satisfy the security requirement described in 4.1. It is possible that individuals may attempt to generate passphrases which do not satisfy this requirement while operating under the false sense of security gained through a system-generated output. We emphasize that satisfying the security requirement is critical to creating good passphrases and urge a considered use of our method. We document how this security requirement is derived and why it is justified. Additionally our code provides a direct way to calculating the security metric of entropy in each output.

References

Avast. 2024. Random password generator: Create strong and secure passwords. https://www.avast.com/en-us/random-password-generator#pc. Accessed: 2024-10-15.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: A simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

EF. 2024. Resources for learning english: 3000 most common words in english. https://www.ef.edu/english-resources/english-vocabulary/top-3000-words. Accessed: 2024-10-15.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Richard Hudson. 1994. About 37 percent of word-tokens are nouns. *Language*, 70(2):331–339.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Min Jin, Junbin Ye, Rongxuan Shen, and Huaxing Lu. 2024. Search-based ordered password generation of autoregressive neural networks. *arXiv preprint arXiv:2403.09954*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *CoRR*, abs/2001.08361.

Saranga Komanduri, Richard Shay, Patrick Gage Kelley, Michelle L Mazurek, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Serge Egelman. 2011. Of passwords and people: measuring the effect of password-composition policies. In *Proceedings of the sigchi conference on human factors in computing systems*, pages 2595–2604.

Yair Lakretz, Théo Desbordes, Dieuwke Hupkes, and Stanislas Dehaene. 2022. Can transformers process recursive nested constructions, like humans? In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3226–3232.

Andrew Lampinen. 2024. Can language models handle recursively nested grammatical structures? a case study on comparing models and humans. *Computational Linguistics*, pages 1–35.

Kevin Lee, Sten Sjöberg, and Arvind Narayanan. 2022. Password policies of most top websites fail to follow best practices. In *Eighteenth Symposium on Usable Privacy and Security (SOUPS 2022)*, pages 561–580, Boston, MA. USENIX Association.

Rui Meng, Xingdi Yuan, Tong Wang, Sanqiang Zhao, Adam Trischler, and Daqing He. 2021. An empirical study on neural keyphrase generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4985–5007.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*.

Guillaume Sanchez, Honglu Fan, Alexander Spangher, Elad Levi, Pawan Sasanka Ammanamanchi, and Stella Biderman. 2023. Stay on topic with classifier-free guidance. *arXiv preprint arXiv:2306.17806*.

Claude Elwood Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Richard Shay, Saranga Komanduri, Adam L Durity, Phillip Huh, Michelle L Mazurek, Sean M Segreti, Blase Ur, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. 2016. Designing password policies for strength and usability. *ACM Transactions on Information and System Security (TISSEC)*, 18(4):1–34.

Joshua Tan, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. 2020a. Practical recommendations for stronger, more usable passwords combining minimum-strength, minimum-length, and blocklist requirements. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 1407–1426.

Joshua Tan, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. 2020b. Practical recommendations for stronger, more usable passwords combining minimum-strength, minimum-length, and blocklist requirements. CCS '20, page 1407–1426, New York, NY, USA. Association for Computing Machinery.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Bailin Wang, Zi Wang, Xuezhi Wang, Yuan Cao, Rif A Saurous, and Yoon Kim. 2024. Grammar prompting for domain-specific language generation with large language models. *Advances in Neural Information Processing Systems*, 36.

Ding Wang, Haibo Cheng, Ping Wang, Xinyi Huang, and Gaopeng Jian. 2017. Zipf's law in passwords. *IEEE Transactions on Information Forensics and Security*, 12(11):2776–2791.

Ding Wang, Yunkai Zou, Yuan-An Xiao, Siqi Ma, and Xiaofeng Chen. 2023. {Pass2Edit}: A {Multi-Step} generative model for guessing edited passwords. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 983–1000.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019a. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019b. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Xiaoyuan Wu, Collins W Munyendo, Eddie Cosic, Genevieve A Flynn, Olivia Legault, and Adam J Aviv. 2022. User perceptions of five-word passwords. In *Proceedings of the 38th Annual Computer Security Applications Conference*, pages 605–618.

Ming Xu, Chuanwang Wang, Jitao Yu, Junjie Zhang, Kai Zhang, and Weili Han. 2021. Chunk-level password guessing: Towards modeling refined password composition representations. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 5–20.

A Appendix

A.1 Models

We generate texts using these chat/instruct models:

- Llama-2-7b-chat-hf (*Llama-2-7B*)
- Llama-2-13b-chat-hf (*Llama-2-13B*)
- Llama-2-70b-chat-hf (*Llama-2-70B*)
- Meta-Llama-3-8B-Instruct (*Llama-3-8B*)
- Meta-Llama-3-70B-Instruct (*Llama-3-70B*)
- Mistral-7B-Instruct-v0.2 (*Mistral-7B*)
- Mixtral-8x7B-Instruct-v0.1 (*Mixtral-8x7B*)
- gemma-7b-it (*gemma-7B*)

The above model names contain the approximate number of parameters in the model as in "7b" or "7B" signifying seven billion parameters. The first three models are part of the Llama-2 family of models (Wolf et al., 2019a) licensed under the Llama 2 Community License. The Llama-3 models (Dubey et al., 2024) are licensed under the Meta Llama 3 Community License. The Mistral (Jiang et al., 2023) and Mixtral (Jiang et al., 2024) models are licensed under the Mistral AI non-production license. The gemma model (Team et al., 2024) is licensed under Gemma Terms of Use. Our use of these models have complied with the relevant license agreement and have been for scientific and non-commercial purposes.

A.2 Scientific Artifacts

The above models are publicly available from HuggingFace at the specific model page of https://huggingface.co/ using the transformers library (Wolf et al., 2019b). Up to six NVIDIA RTX A5000 graphics cards may be needed to generate texts using these models with generation of each batch taking a few minutes, with most of the time dedicated to loading the model. The smaller models and quantization of larger models allow fewer graphics cards to be used.

A.3 Direct-question prompt versus template prompt

The direct-question prompts sometimes lead to outputs that start with filler words such as “Sure", “Certainly", “Here", “Here’s", “Okay", “Ok". As discussed in A.3, this seems to occur as there is no guidance on the desired output form in the direct-question prompts and the direct-question prompts evoke a more conversational response. The template prompt effectively eliminates the occurrence of these extra words. See Figure .

model	base	story	passp	sum	TLDR
Llama-2-7B	0.0	0.1	0.4	0.5	0.4
Llama-2-13B	0.0	0.6	0.7	0.9	0.9
Llama-2-70B	0.0	0.2	0.1	0.0	0.1
Llama-3-8B	0.0	0.0	0.3	0.0	0.0
Llama-3-70B	0.0	0.0	0.2	0.0	0.0
Mistral-7B	0.0	0.0	0.0	0.0	0.0
Mixtral-8x7B	0.0	0.0	0.0	0.0	0.0
gemma-7B	0.0	0.0	1.0	1.0	1.0

Table 10: Proportion of outputs that start with “Sure", “Certainly", “Here", “Here’s", “Okay", or “Ok", sampled with temperature of 1.4 and top-p of 0.95 for the base prompt and these direct-question prompts. “story" column corresponds to prompt of “Give me a story in six words"; “passp" to “Write a passphrase in six words"; “sum" to “Write a summary of a story in six words"; “TLDR" to “Write a tldr of a story in six words"

The difficulty of using a direct-question prompt to achieve all three passphrase criteria as discussed in A.3 and seen in Tables 1, 11 and 12 is generally true and can be observed at other temperature and top-p parameters. For example, see Tables 13, 14, and 15.

A.4 Increasing temperature or top-p generally increases entropy

Note the effect of increased entropy as temperature or top-p increases, but the other passphrase criteria of limited number of words or all words being correctly spelt English words may suffer. See the statistics on output generated using the base prompt at temperatures of 1.0, 1.2, and 1.4 and top-p of 0.95, 0.99, and 1.0 in Tables 16 to 24.

A.5 Steering towards a pre-specified topic

See Table 25 for prevalence of science and science-related words in outputs of prompts that include science and science related topics, in comparison to the base prompt. As expected science-prompt had outputs with more science or science-related words.

model	story	passp	sum	TLDR
Llama-2-7B	0.9	0.6	0.5	0.5
Llama-2-13B	0.4	0.3	0.1	0.1
Llama-2-70B	0.8	0.9	0.9	0.9
Llama-3-8B	1.0	0.7	1.0	1.0
Llama-3-70B	1.0	0.8	1.0	1.0
Mistral-7B	0.0	0.1	0.1	0.0
Mixtral-8x7B	0.2	0.1	0.1	0.1
gemma-7B	0.0	0.1	0.0	0.0

Table 11: For direction-question prompts, proportion of outputs that are of eight or fewer words. "story" column corresponds to prompt of "Give me a story in six words"; "passp" to "Write a passphrase in six words"; "sum" to "Write a summary of a story in six words"; "TLDR" to "Write a tldr of a story in six words" (sampled with temperature of 1.4 and top-p of 0.95)

.

model	story	passp	sum	TLDR
Llama-2-7B	1.0	0.4	1.0	0.5
Llama-2-13B	0.9	0.2	0.8	0.3
Llama-2-70B	0.9	0.7	0.7	0.6
Llama-3-8B	0.9	0.4	0.9	0.8
Llama-3-70B	1.0	0.7	1.0	1.0
Mistral-7B	0.5	0.2	0.5	0.6
Mixtral-8x7B	0.6	0.1	0.4	0.3
gemma-7B	1.0	0.0	1.0	0.1

Table 12: For direction-question prompts, proportion of outputs that are of entirely English words. "story" column corresponds to prompt of "Give me a story in six words"; "passp" to "Write a passphrase in six words"; "sum" to "Write a summary of a story in six words"; "TLDR" to "Write a tldr of a story in six words" (sampled with temperature of 1.4 and top-p of 0.95)

.

model	story	passp	sum	TLDR
Llama-2-7B	0.0	0.0	0.0	0.0
Llama-2-13B	0.0	0.0	0.0	0.0
Llama-2-70B	0.0	0.0	0.0	0.0
Llama-3-8B	0.0	0.1	0.0	0.0
Llama-3-70B	0.0	0.0	0.0	0.0
Mistral-7B	0.1	0.5	0.0	0.0
Mixtral-8x7B	0.0	0.0	0.0	0.0
gemma-7B	0.0	0.0	0.0	0.0

Table 13: Proportion of outputs that have entropy $\geq$ 47, sampled with temperature of 1.0 and top-p of 0.95 for different prompts. “story" column corresponds to prompt of “Give me a story in six words"; “passp" to “Write a passphrase in six words"; “sum" to “Write a summary of a story in six words"; “TLDR" to “Write a tldr of a story in six words"

model	story	passp	sum	TLDR
Llama-2-7B	0.9	0.2	0.2	0.2
Llama-2-13B	0.1	0.2	0.0	0.0
Llama-2-70B	0.8	1.0	1.0	0.9
Llama-3-8B	1.0	0.7	1.0	1.0
Llama-3-70B	1.0	0.7	1.0	1.0
Mistral-7B	0.0	0.1	0.0	0.0
Mixtral-8x7B	0.0	0.0	0.2	0.0
gemma-7B	0.0	0.1	0.0	0.0

Table 14: Proportion of outputs that are of eight or fewer words, sampled with temperature of 1.0 and top-p of 0.95 for different prompts. “story" column corresponds to prompt of “Give me a story in six words"; “passp" to “Write a passphrase in six words"; “sum" to “Write a summary of a story in six words"; “TLDR" to “Write a tldr of a story in six words"

model	story	passp	sum	TLDR
Llama-2-7B	1.0	0.2	1.0	0.2
Llama-2-13B	1.0	0.2	0.8	0.2
Llama-2-70B	1.0	0.9	1.0	0.8
Llama-3-8B	1.0	0.2	1.0	1.0
Llama-3-70B	1.0	0.7	1.0	1.0
Mistral-7B	0.9	0.5	0.7	0.9
Mixtral-8x7B	0.6	0.0	0.4	0.2
gemma-7B	1.0	0.0	1.0	0.0

Table 15: Proportion of outputs that are of entirely English words, sampled with temperature of 1.0 and top-p of 0.95 for different prompts. “story" column corresponds to prompt of “Give me a story in six words"; “passp" to “Write a passphrase in six words"; “sum" to “Write a summary of a story in six words"; “TLDR" to “Write a tldr of a story in six words"

model	entropy $\geq$ 47.0	num words ≤ 8	Eng	All criteria
Llama-2-7B	0.1	1.0	1.0	0.1
Llama-2-13B	0.1	1.0	1.0	0.1
Llama-2-70B	0.0	1.0	0.9	0.0
Llama-3-8B	0.0	1.0	1.0	0.0
Llama-3-70B	0.0	1.0	1.0	0.0
Mistral-7B	0.5	1.0	0.9	0.4
Mixtral-8x7B	0.5	1.0	0.8	0.3
gemma-7B	0.1	0.9	0.9	0.0

Table 16: Proportion of output that meet criteria of entropy ≥ 47.0, num_words ≤ 8 and consisting of all English words, separately, and all criteria concurrently using the base prompt, temperature 1.0 and top-p of 0.95

model	entropy $\geq$ 47.0	num words ≤ 8	Eng	All criteria
Llama-2-7B	0.2	1.0	1.0	0.2
Llama-2-13B	0.2	1.0	0.9	0.2
Llama-2-70B	0.1	1.0	0.9	0.1
Llama-3-8B	0.1	1.0	1.0	0.1
Llama-3-70B	0.0	1.0	1.0	0.0
Mistral-7B	0.6	0.9	0.9	0.5
Mixtral-8x7B	0.6	1.0	0.8	0.5
gemma-7B	0.2	0.9	0.9	0.1

Table 17: Proportion of output that meet criteria of entropy ≥ 47.0, num_words ≤ 8 and consisting of all English words, separately, and all criteria concurrently using the base prompt, temperature 1.0 and top-p of 0.99.

model	entropy $\geq$ 47.0	num words ≤ 8	Eng	All criteria
Llama-2-7B	0.2	1.0	0.9	0.2
Llama-2-13B	0.2	1.0	0.9	0.2
Llama-2-70B	0.1	1.0	0.9	0.1
Llama-3-8B	0.1	1.0	0.9	0.1
Llama-3-70B	0.0	1.0	1.0	0.0
Mistral-7B	0.7	0.9	0.9	0.5
Mixtral-8x7B	0.6	1.0	0.9	0.5
gemma-7B	0.2	0.9	0.9	0.1

Table 18: Proportion of output that meet criteria of entropy ≥ 47.0, num_words ≤ 8 and consisting of all English words, separately, and all criteria concurrently using the base prompt, temperature 1.0 and top-p of 1.0

model	entropy $\geq$ 47.0	num words ≤ 8	Eng	All criteria
Llama-2-7B	0.4	1.0	0.9	0.4
Llama-2-13B	0.5	1.0	0.9	0.5
Llama-2-70B	0.2	1.0	0.8	0.1
Llama-3-8B	0.4	1.0	0.9	0.4
Llama-3-70B	0.0	1.0	1.0	0.0
Mistral-7B	0.8	0.9	0.9	0.7
Mixtral-8x7B	0.9	1.0	0.8	0.7
gemma-7B	0.6	0.8	0.7	0.2

Table 19: Proportion of output that meet criteria of entropy ≥ 47.0, num_words ≤ 8 and consisting of all English words, separately, and all criteria concurrently using the base prompt, temperature 1.2 and top-p of 0.95

model	entropy $\geq$ 47.0	num words ≤ 8	Eng	All criteria
Llama-2-7B	0.5	1.0	0.9	0.5
Llama-2-13B	0.6	1.0	0.8	0.5
Llama-2-70B	0.4	1.0	0.7	0.2
Llama-3-8B	0.5	1.0	0.9	0.4
Llama-3-70B	0.1	1.0	0.9	0.1
Mistral-7B	0.9	0.9	0.8	0.7
Mixtral-8x7B	0.9	0.9	0.8	0.7
gemma-7B	0.7	0.8	0.7	0.3

Table 20: Proportion of output that meet criteria of entropy ≥ 47.0, num_words ≤ 8 and consisting of all English words, separately, and all criteria concurrently using the base prompt, temperature 1.2 and top-p of 0.99

model	entropy $\geq$ 47.0	num words ≤ 8	Eng	All criteria
Llama-2-7B	0.6	1.0	0.9	0.5
Llama-2-13B	0.7	1.0	0.8	0.5
Llama-2-70B	0.4	0.9	0.7	0.2
Llama-3-8B	0.6	1.0	0.8	0.4
Llama-3-70B	0.2	1.0	0.9	0.1
Mistral-7B	0.9	0.8	0.8	0.6
Mixtral-8x7B	0.9	0.9	0.8	0.7
gemma-7B	0.7	0.8	0.7	0.3

Table 21: Proportion of output that meet criteria of entropy ≥ 47.0, num_words ≤ 8 and consisting of all English words, separately, and all criteria concurrently using the base prompt, temperature 1.2 and top-p of 1.0

model	entropy $\geq$ 47.0	num words ≤ 8	Eng	All criteria
Llama-2-7B	0.8	1.0	0.8	0.6
Llama-2-13B	0.9	1.0	0.8	0.7
Llama-2-70B	0.7	0.9	0.6	0.3
Llama-3-8B	0.9	1.0	0.6	0.6
Llama-3-70B	0.3	1.0	0.8	0.1
Mistral-7B	1.0	0.7	0.6	0.6
Mixtral-8x7B	1.0	0.9	0.7	0.7
gemma-7B	0.9	0.7	0.5	0.3

Table 22: Proportion of output that meet criteria of entropy ≥ 47.0, num_words ≤ 8 and consisting of all English words, separately, and all criteria concurrently using the base prompt, temperature 1.4 and top-p of 0.95

model	entropy $\geq$ 47.0	num words ≤ 8	Eng	All criteria
Llama-2-7B	0.8	0.9	0.7	0.5
Llama-2-13B	0.9	1.0	0.7	0.6
Llama-2-70B	0.8	0.8	0.5	0.3
Llama-3-8B	0.9	1.0	0.5	0.4
Llama-3-70B	0.5	1.0	0.6	0.2
Mistral-7B	1.0	0.6	0.5	0.4
Mixtral-8x7B	1.0	0.9	0.7	0.6
gemma-7B	1.0	0.7	0.4	0.3

Table 23: Proportion of output that meet criteria of entropy ≥ 47.0, num_words ≤ 8 and consisting of all English words, separately, and all criteria concurrently using the base prompt, temperature 1.4 and top-p of 0.99

model	entropy $\geq$ 47.0	num words ≤ 8	Eng	All criteria
Llama-2-7B	0.9	0.9	0.6	0.4
Llama-2-13B	0.9	1.0	0.6	0.6
Llama-2-70B	0.8	0.9	0.5	0.3
Llama-3-8B	0.9	1.0	0.5	0.4
Llama-3-70B	0.5	1.0	0.6	0.1
Mistral-7B	1.0	0.6	0.4	0.3
Mixtral-8x7B	1.0	0.9	0.7	0.7
gemma-7B	1.0	0.7	0.4	0.3

Table 24: Proportion of output that meet criteria of entropy ≥ 47.0, num_words ≤ 8 and consisting of all English words, separately, and all criteria concurrently using the base prompt, temperature 1.4 and top-p of 1.0

model	base prompt science output	base prompt sci-rel output	science prompt science output	science prompt sci-rel output
Llama-2-7B	0.01	0.05	0.50	0.05
Llama-2-13B	0.01	0.02	0.17	0.09
Llama-2-70B	0.02	0.05	0.98	0.13
Llama-3-8B	0.00	0.05	0.12	0.07
Llama-3-70B	0.01	0.17	0.12	0.25
Mistral-7B	0.00	0.01	0.08	0.05
Mixtral-8x7B	0.01	0.03	0.08	0.07
gemma-7B	0.01	0.05	0.09	0.05

Table 25: Proportion of output that contains “science"/“scientist"/“scientific" and proportion of output that contains science-related words or word-stems (“robot", “innovat", “discover", “engineer" or “experiment"), compared between base prompt and science-prompt.

A.6 High entropy, low entropy

Tables 26, 27, and 28 are the average output entropy for base prompt, low-entropy prompt, and high-entropy prompt at top-p of 0.95 and at temperatures of 1.0, 1.2 and 1.4, respectively. Note the general trend of higher entropy prompt promoting higher entropy output but the scale of the effect varies for different temperatures.

A.7 First token entropy

In Section 6.1, we focus on the first token and calculate the expected entropy of the first token at various temperatures by summing over each poten-

model	base prompt	low-ent prompt	high-ent prompt
Llama-2-7B	31.5	16.9	29.6
Llama-2-13B	34.8	16.6	31.8
Llama-2-70B	22.6	13.9	19.4
Llama-3-8B	31.9	19.3	30.0
Llama-3-70B	15.7	3.4	16.0
Mistral-7B	47.0	21.2	45.6
Mixtral-8x7B	49.2	27.6	55.1
gemma-7B	32.6	23.5	27.1

Table 26: Average entropy of output using the base prompt, prompt containing low entropy examples and prompt containing high entropy examples, with temperature of 1.0 and top-p of 0.95.

model	base prompt	low-ent prompt	high-ent prompt
Llama-2-7B	46.3	25.2	45.2
Llama-2-13B	47.8	24.1	41.3
Llama-2-70B	39.7	30.5	45.4
Llama-3-8B	46.3	27.4	43.3
Llama-3-70B	24.5	6.8	23.7
Mistral-7B	64.5	32.1	62.8
Mixtral-8x7B	59.7	37.1	66.2
gemma-7B	54.3	37.9	44.3

Table 27: Average entropy of output using the base prompt, prompt containing low entropy examples and prompt containing high entropy examples, with temperature of 1.2 and top-p of 0.95.

model	base prompt	low-ent prompt	high-ent prompt
Llama-2-7B	64.4	44.3	66.5
Llama-2-13B	61.8	44.4	74.6
Llama-2-70B	106.0	87.5	138.3
Llama-3-8B	69.7	50.3	61.0
Llama-3-70B	44.8	16.0	41.9
Mistral-7B	119.8	59.7	111.9
Mixtral-8x7B	73.3	55.3	93.1
gemma-7B	85.1	65.7	69.1

Table 28: Average entropy of output using the base prompt, prompt containing low entropy examples and prompt containing high entropy examples, with temperature of 1.4 and top-p of 0.95.

tial token's entropy weighted by its probability of being selected as the first token. To make this more concrete, see Figure 4, which plots the 50 tokens with the highest probability (corresponding to the lowest entropy) of being selected as the first token using the base-prompt at temperature 1.0 and top-p of 1.0 in the Gemma-7B model.

A.8 Additional parameter settings for passphrase generation

In addition to the parameter settings listed in Table 9, the parameters in Tables 29, 30, and 31 can also be used to generate good passphrases.

A.9 Logits adjustments in passphrase generation

In order to standardize the output of generated passphrases, tokens containing non-ascii, digit and certain punctuation characters have their probabilities zeroed out. Additionally tokens containing the newline, exclamation, period and question mark are replaced by the eos token and no further token is sampled in the sequence.

A.10 User Study

The users are recruited from a university setting and are university students. They responded to requests for volunteers for a user study involving passphrases. No payment is offered in exchange for participation. No personal information about any individual participant has been collected. Only their response to the survey questions about their preference, opinions, and ability to recall passphrases has been collected.

The relevant experts has determined the user study described in this paper is exempt form Institutional Review Board review and has reviewed the information provided to users. The users has been notified in writing that their participation is entirely voluntary, that the risks are comparable to those encountered in everyday life, and that none of their personal data would be collected.

model	temp	top-p	q-trunc	all-criteria
Llama-2-13B	1.20	0.95	0	0.32
Llama-2-13B	1.20	0.99	0	0.40
Llama-2-13B	1.20	1.00	0	0.41
Llama-2-13B	1.40	0.95	0	0.56
Llama-2-13B	1.40	0.99	0	0.59
Llama-2-13B	1.40	1.00	0	0.56
Llama-2-70B	1.40	0.95	0	0.30
Llama-2-70B	1.40	0.99	0	0.30
Llama-2-70B	1.40	1.00	0	0.31
Llama-2-7B	1.20	0.99	0	0.29
Llama-2-7B	1.20	1.00	0	0.29
Llama-2-7B	1.40	0.95	0	0.48
Llama-2-7B	1.40	0.99	0	0.39
Llama-2-7B	1.40	1.00	0	0.39
Llama-3-8B	1.20	0.95	0	0.38
Llama-3-8B	1.20	0.99	0	0.42
Llama-3-8B	1.20	1.00	0	0.44
Llama-3-8B	1.40	0.95	0	0.55
Llama-3-8B	1.40	0.99	0	0.53
Llama-3-8B	1.40	1.00	0	0.52
Mistral-7B	1.00	0.99	0	0.36
Mistral-7B	1.00	1.00	0	0.41
Mistral-7B	1.20	0.95	0	0.52
Mistral-7B	1.20	0.99	0	0.57
Mistral-7B	1.20	1.00	0	0.55
Mistral-7B	1.40	0.95	0	0.59
Mistral-7B	1.40	0.99	0	0.48
Mistral-7B	1.40	1.00	0	0.42
Mixtral-8x7B	1.00	0.95	0	0.37
Mixtral-8x7B	1.00	0.99	0	0.48
Mixtral-8x7B	1.00	1.00	0	0.55
Mixtral-8x7B	1.20	0.95	0	0.68
Mixtral-8x7B	1.20	0.99	0	0.63
Mixtral-8x7B	1.20	1.00	0	0.63
Mixtral-8x7B	1.40	0.95	0	0.56
Mixtral-8x7B	1.40	0.99	0	0.52
Mixtral-8x7B	1.40	1.00	0	0.48
gemma-7B	1.20	0.99	0	0.30
gemma-7B	1.20	1.00	0	0.30
gemma-7B	1.40	0.95	0	0.38
gemma-7B	1.40	0.99	0	0.36
gemma-7B	1.40	1.00	0	0.34

Table 29: Settings for passphrase generation with q-trunc of 0. For each model, use the base-prompt and set temperature, top-p, and q-truncation as in the above to create good passphrases. The last column lists the proportion of output which meets all three criteria: entropy ≥ 49, number of words ≤ 8, and composed of English words.

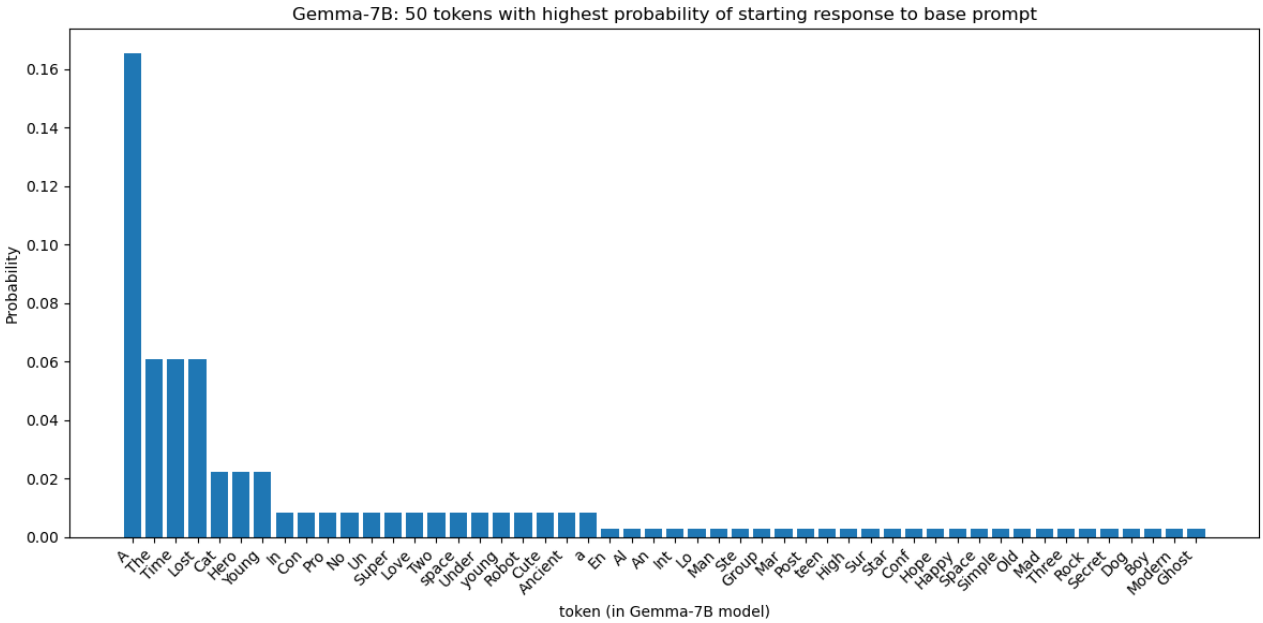

Figure 4: Gemma-7B model: Potential first tokens and their probability (top 50 tokens pictured).

model	temp	top-p	q-trunc	all-criteria
Llama-2-13B	1.20	0.95	0.20	0.45
Llama-2-13B	1.20	0.99	0.20	0.52
Llama-2-13B	1.20	1.00	0.20	0.56
Llama-2-13B	1.40	0.95	0.20	0.70
Llama-2-13B	1.40	0.99	0.20	0.61
Llama-2-13B	1.40	1.00	0.20	0.56
Llama-2-70B	1.40	0.95	0.20	0.38
Llama-2-70B	1.40	0.99	0.20	0.30
Llama-2-70B	1.40	1.00	0.20	0.31
Llama-2-7B	1.20	0.95	0.20	0.39
Llama-2-7B	1.20	0.99	0.20	0.49
Llama-2-7B	1.20	1.00	0.20	0.45
Llama-2-7B	1.40	0.95	0.20	0.60
Llama-2-7B	1.40	0.99	0.20	0.48
Llama-2-7B	1.40	1.00	0.20	0.43
Llama-3-8B	1.20	0.95	0.20	0.42
Llama-3-8B	1.20	0.99	0.20	0.48
Llama-3-8B	1.20	1.00	0.20	0.48
Llama-3-8B	1.40	0.95	0.20	0.58
Llama-3-8B	1.40	0.99	0.20	0.49
Llama-3-8B	1.40	1.00	0.20	0.45
Mistral-7B	1.00	0.95	0.20	0.34
Mistral-7B	1.00	0.99	0.20	0.43
Mistral-7B	1.00	1.00	0.20	0.45
Mistral-7B	1.20	0.95	0.20	0.66
Mistral-7B	1.20	0.99	0.20	0.66
Mistral-7B	1.20	1.00	0.20	0.59
Mistral-7B	1.40	0.95	0.20	0.59
Mistral-7B	1.40	0.99	0.20	0.45
Mistral-7B	1.40	1.00	0.20	0.41
Mixtral-8x7B	1.00	0.95	0.20	0.43
Mixtral-8x7B	1.00	0.99	0.20	0.53
Mixtral-8x7B	1.00	1.00	0.20	0.62
Mixtral-8x7B	1.20	0.95	0.20	0.66
Mixtral-8x7B	1.20	0.99	0.20	0.71
Mixtral-8x7B	1.20	1.00	0.20	0.61
Mixtral-8x7B	1.40	0.95	0.20	0.52
Mixtral-8x7B	1.40	0.99	0.20	0.45
Mixtral-8x7B	1.40	1.00	0.20	0.41
gemma-7B	1.20	0.95	0.20	0.27
gemma-7B	1.20	0.99	0.20	0.33
gemma-7B	1.20	1.00	0.20	0.32
gemma-7B	1.40	0.95	0.20	0.35
gemma-7B	1.40	0.99	0.20	0.30
gemma-7B	1.40	1.00	0.20	0.30

Table 30: Settings for passphrase generation with q-trunc of 0.20. For each model, use the base-prompt and set temperature, top-p, and q-truncation as in the above to create good passphrases. The last column lists the proportion of output which meets all three criteria: entropy ≥ 49, number of words ≤ 8, and composed of English words.

model	temp	top-p	q-trunc	all-criteria
Llama-2-13B	1.00	1.00	0.35	0.29
Llama-2-13B	1.20	0.95	0.35	0.57
Llama-2-13B	1.20	0.99	0.35	0.62
Llama-2-13B	1.20	1.00	0.35	0.62
Llama-2-13B	1.40	0.95	0.35	0.56
Llama-2-70B	1.20	0.99	0.35	0.27
Llama-2-70B	1.20	1.00	0.35	0.27
Llama-2-7B	1.00	1.00	0.35	0.28
Llama-2-7B	1.20	0.95	0.35	0.48
Llama-2-7B	1.20	0.99	0.35	0.49
Llama-2-7B	1.20	1.00	0.35	0.48
Llama-2-7B	1.40	0.95	0.35	0.57
Llama-3-70B	1.40	1.00	0.35	0.27
Llama-3-8B	1.00	0.99	0.35	0.26
Llama-3-8B	1.00	1.00	0.35	0.30
Llama-3-8B	1.20	0.95	0.35	0.52
Llama-3-8B	1.20	0.99	0.35	0.54
Llama-3-8B	1.20	1.00	0.35	0.53
Llama-3-8B	1.40	0.95	0.35	0.59
Llama-3-8B	1.40	0.99	0.35	0.41
Llama-3-8B	1.40	1.00	0.35	0.36
Mistral-7B	1.00	0.95	0.35	0.36
Mistral-7B	1.00	0.99	0.35	0.53
Mistral-7B	1.00	1.00	0.35	0.58
Mistral-7B	1.20	0.95	0.35	0.65
Mistral-7B	1.20	0.99	0.35	0.59
Mistral-7B	1.20	1.00	0.35	0.56
Mistral-7B	1.40	0.95	0.35	0.55
Mistral-7B	1.40	0.99	0.35	0.38
Mistral-7B	1.40	1.00	0.35	0.34
Mixtral-8x7B	1.00	0.95	0.35	0.48
Mixtral-8x7B	1.00	0.99	0.35	0.58
Mixtral-8x7B	1.00	1.00	0.35	0.67
Mixtral-8x7B	1.20	0.95	0.35	0.66
Mixtral-8x7B	1.20	0.99	0.35	0.50
Mixtral-8x7B	1.20	1.00	0.35	0.49
Mixtral-8x7B	1.40	0.95	0.35	0.48
Mixtral-8x7B	1.40	0.99	0.35	0.37
gemma-7B	1.20	0.95	0.35	0.41
gemma-7B	1.20	0.99	0.35	0.42
gemma-7B	1.20	1.00	0.35	0.46
gemma-7B	1.40	0.95	0.35	0.28

Table 31: Settings for passphrase generation with q-trunc of 0.35. For each model, use the base-prompt and set temperature, top-p, and q-truncation as in the above to create good passphrases. The last column lists the proportion of output which meets all three criteria: entropy ≥ 49, number of words ≤ 8, and composed of English words.