Beyond the Spelling Miracle: Investigating Substring Awareness in Character-Blind Language Models

Cristiano Ciaccio, Marta Sartor, Alessio Miaschi, Felice Dell'Orletta
Istituto di Linguistica Computazionale "Antonio Zampolli" (CNR-ILC)
ItaliaNLP Lab, Pisa
{name.surname}@ilc.cnr.it

Abstract

Correctly identifying characters and substrings of words should be a basic but essential ability of any Language Model that aims to proficiently understand and produce language. Despite so, the majority of Pre-trained Language Models (PLMs) are "character-blind" and struggle in spelling tasks, although they still seem to acquire some character knowledge during pre-training, a phenomenon dubbed *Spelling Miracle*. To shed light on this phenomenon, we systematically evaluate a range of PLMs with different parameter sizes using a controlled binary substring identification task. Through a series of experiments, we propose the first comprehensive investigation on where, when, and how PLMs develop awareness of characters and substrings, with a particular linguistic focus on morphemic units such as prefixes, suffixes, and roots.

1 Introduction

Current Pre-trained Language Models (PLMs) are trained over large corpora of text leveraging the Causal Language Modeling task (Bengio et al., 2003): predicting the next token in a sequence of tokens, where the criteria by which words are split into tokens is obtained statistically using compression algorithms such as Byte Pair Encoding (BPE) (Sennrich et al., 2016). This approach produces a segmentation that encourages the encoding of high-frequency words into single tokens and splits low-frequency words into frequent subtokens. This process results in models that are "**character-blind**": they are not aware of characters and their pre-training objective is not designed to learn representations of this knowledge. Despite this, recent studies have revealed that PLMs exhibit limited spelling abilities, a phenomenon dubbed the *Spelling Miracle* (Liu et al., 2023).

Yet, the ability to identify characters and substrings within words is apparently trivial but it's fundamental to robust language understanding. Such knowledge is crucial not only in tasks that require character information (spelling, solving typos, rhymes generation, etc.) but also in scenarios where the understanding of a word, and therefore the full meaning of a sentence, is strictly related to the parts of which a word is composed. For instance, in the sentence "this device can decaffeinify any drink," the neologism *decaffeinify* might be tokenized into ['dec', 'affe', 'in', 'ify'][1]. While the model has an explicit representation for the *-ify* suffix that can encode the meaning shift caused by this morpheme, both the prefix *de-* and the root *caffe* must be reconstructed via implicit character-level knowledge. Due to the compositional nature and the high productivity of language (60 % of the new words a reader will encounter are morphologically complex, Angelelli et al., 2014), PLMs are often faced with these challenges and, as shown by Zheng et al. (2024), downstream performance severely degrades with sentences involving morphological neologisms.

Despite these observations, the emergence of character and substring knowledge in character-blind models remains an underlooked topic, particularly from a linguistic perspective. Moreover, existing studies relied on training probing classifiers (Kaushal and Mahowald, 2022; Itzhak and Levy, 2022) or zero/few-shots prompting (Edman et al., 2024) which have limitations (Belinkov, 2022; Laskar et al., 2024). To address this gap, we propose an experimental setup that does not rely on probing or prompting, and we extend current studies by analyzing how PLMs identify substrings of various lengths, at various positions, when this competence emerges during pre-training and how performance changes according to linguistic variables such as the substring being a morpheme (i.e.,

[1]The reported tokenization is performed by the Pythia tokenizer: https://huggingface.co/EleutherAI/pythia-1B.

Findings of the Association for Computational Linguistics: ACL 2025, pages 11361–11372

a prefix, suffix or a root). Specifically, we selected decoder-only models from the Pythia family (Biderman et al., 2023) of different parameter sizes, leveraging the MorphoLex database (Sánchez-Gutiérrez et al., 2018) to construct our dataset.

Our main research questions are: *(i)* Do character-blind models have an understanding of the character composition of words? *(ii)* How does this competence change according to model size, substring length and position? *(iii)* When does this competence emerge during pre-training? *(iv)* Are morphemes recognized the most? If so, which ones? *(v)* Are there linguistic and non-linguistic variables that affect this competence?

2 Related Work

Assessing the character-level knowledge of PLMs started to emerge recently as a research area. Despite so, few works target this phenomenon explicitly, often omitting the related linguistic aspects. Itzhak and Levy (2022) probed the embedding matrix of PLMs to reconstruct complete word spellings, finding that, curiously, GPT2-medium and RoBERTa-large solved the task correctly 30% of the time. Kaushal and Mahowald (2022) trained probing classifiers to predict the presence or absence of a particular character/substring in a token, finding that larger models store more character knowledge. Edman et al. (2024), Efrat et al. (2023), Huang et al. (2023) and Suvarna et al. (2024) proposed suites of tasks to evaluate PLMs that involve character-related knowledge, ranging from syllable counting and rhyme word generation to inverse spelling and misspelling correction. From a morphological perspective, Lerner and Yvon (2025) studied the ability of PLMs to perform affixation, relating it to the generation of nonce words, finding that prefixation is harder due to tokenization artifacts.

Differently from the aforementioned works, our approach focuses specifically on character and substring-level identification across different lengths and positions, focusing also on linguistically meaningful units like prefixes, suffixes and roots, with a particular emphasis on when and how such knowledge develops throughout pre-training.

3 Our Approach

To assess a LLM's knowledge of the character composition of its tokens and the sensitivity towards sublexical morphemes, we developed a minimal fine-tuning approach using a character-n-gram binary classification task: given a word *w* and a substring *n-gram* that may or may not appear in *w*, the source text is "Is *n-gram* inside *w*?", the expected output is {*yes*, *no*}. The *n-gram* length varies between 1 (a single character) and 6 and can be located at the start, the middle and the end of *w*. This method produces 18 classes (6 n-grams lengths times 3 positions). Relying on the MorphoLex dataset, we kept an important distinction between derived words, i.e. words formed via morphological processes (*re* + *load*), and morphologically simple words, i.e. words with no meaningful sublexical structure (*load*), and whether the asked n-gram is a morpheme, i.e. a prefix ("*dis*-like"), a suffix ("like-*ly*"), a root ("un-*like*-ly"), or a meaningless substring ("*l*-ike"). This distinction allows us to measure performance for each of the 18 classes while also assessing the change in performance between morphemic and non-morphemic n-grams.

To ensure we are measuring the model's existing character-level knowledge rather than injecting it, each model is fine-tuned on a small, controlled and balanced set of examples (both positive and negative), which is just sufficient to teach the task format. On the other hand, the test set is comprehensive, designed to extensively evaluate several axes of character-related competence.

Since our focus is on the models' competence – specifically, their ability to store and access character-level knowledge – rather than their generative performance (Hu and Levy, 2023), fine-tuning provides a controlled way to target and elicit the specific knowledge we aim to investigate. To test whether the understanding of the character composition of tokens comes from the models' pre-training phase or from our fine-tuning setup, we extend these experiments to the pre-training checkpoints of the Pythia family, including random initialized weights. This strategy provides a robust baseline and allows us to examine when and how this knowledge emerges during pre-training, as well as how morphological variables affect the acquisition of this competence.

3.1 Dataset

To conduct our experiments we leveraged MorphoLex (Sánchez-Gutiérrez et al., 2018), a valuable database of derivational morphological variables for each complex word in the complete English Lexicon Project (ELP, Balota et al., 2007). Mor-

phoLex contains several word forms, each tagged with a specific prefix-root-suffix signature (PRS) and different morphological variables (affix and root frequency, family size, productivity, and so on). Our dataset is built as follows: using the PRS structure, we focus on single-root words, words containing exactly one prefix, words containing exactly one suffix, and, lastly, words containing both a prefix and a suffix. The single root set represents morphologically simple words, the other sets represent morphologically complex words (also called "derived"). By keeping only morphologically transparent words (each morpheme must be fully intact within a word) and removing inflected forms, we obtained 23 639 words, of which 14 200 are morphologically simple and 9439 are derived.

3.2 Models

We conduct our experiments using the Pythia model family (Biderman et al., 2023), a suite of decoder-only transformers of varying sizes pretrained on the Pile corpus (Gao et al., 2020). Specifically, we employ the 70M, 160M, 410M, and 1B models to assess how character- and morpheme-related competence evolves with increasing model size. Importantly, all the models share the same tokenizer trained on the Pile with a vocabulary size of 50 254, allowing for a fair comparison. Furthermore, to investigate the role of pre-training in the development of this competence, we extend our experiments to the following pre-training checkpoints (where 1 step $\approx$ 2 million tokens): step 0 (randomly initialized weights), 1, 4, 16, 32, 512, 1000, 2000, and 3000 (2% of pre-training).

3.3 Experimental Setting

To design a training set that focuses solely on teaching the task without injecting additional knowledge into the models, we randomly extracted a small subset of 200 words, evenly split between derived and non-derived lemmas, from the overall dataset. Importantly, none of them begin with the characters {a, h, o, v}. By excluding these letters, we prevent the models from encountering n-grams in the start position that begin with these characters, ensuring they must generalize and leverage competence acquired during pre-training. Combined with the experiments on randomly initialized weights, this setup ensures the robustness of our experimental setting. The training set is entirely derived from these 200 words. From the remaining dataset, we randomly sample an additional 200 words to construct a small validation set, while the rest is used for testing, resulting in 95% unseen word tokens in the test set. For any given word *w*, we generate character n-grams of incremental length with $n \in [1, 6]$, starting from each character of *w*. To specify positional context, we prepend "_" to n-grams extracted from the beginning of *w*, append "_" to those from the end, and leave n-grams from the middle unmodified. Given this approach, the word *genesis* would result in the following list of n-grams: ['_g', '_ge', '_gen', '_gene', '_genes', '_genesi', 'e', 'en', 'ene', 'enes', 'enesi', 'enesis_', 'n', 'ne', 'nes', 'nesi', 'nesis_', 'e', 'es', 'esi', 'esis_', 's', 'si', 'sis_', 'i', 'is_', 's_'][2], which exhaust every possible n-gram of length between 1 and 6 in *w*, eventually intercepting morphemes.

Tokenization	Substr. Tokens	Identity Flag
["consci", "ously"]	["ously"]	True
["consci", "ous", "ly"]	["ous", "ly"]	True
["consci", "ous", "ly"]	["ously"]	False

Table 1: Identity filtering process, exemplified for start position.

After computing the n-grams, we generate both a positive and negative example for each. Negative examples are created by randomly selecting an n-gram of the same length and position as the corresponding positive example from a list of n-grams extracted from all the words in MorphoLex[3]. This procedure ensures a balanced number of examples for each class. Overall, by limiting the total number of examples for each of the 18 classes to 300 (for training and validation only), we obtained training, validation, and test sets consisting of 5264, 5248, and 1 233 091 examples, respectively.

Importantly, we flag examples where the subtoken(s) of the n-gram is/are fully present in the subtokens of the tokenized word and label them as identity. For example, considering the word "consciously" and the substring "ously", the identity flagging process is as illustrated in Table 1 (here exemplified on end position, but analogous for start and middle). We introduce this distinction because these instances do not require the model to utilize character-level knowledge but instead to simply apply an identity function. Moreover, tokens starting with "Ġ" (encoded space) are considered an identity with their non-Ġ counterpart. E.g., if *liking* is tokenized as ["Ġlik", "ing"] and the queried substring

[2]We excluded the cases where the n-gram is the word itself.

[3]The sets of n-grams used for generating negative examples are computed separately for the training, validation, and test datasets.

Pythia	Overall	Non-derived	Derived N.M.N.	Derived M.N.
70M	0.62	0.62	0.61	0.74
160M	0.75	0.78	0.73	0.87
410M	0.77	0.79	0.75	0.85
1B	**0.83**	**0.85**	**0.81**	**0.91**

Table 1: Overall F1-Score across model sizes and morphological value.

lik is ["lik"] we would flag this example as identity. This choice is motivated by the work of Lerner and Yvon (2025) in which they found that word-initial and word internal tokens (e.g. "Ġlik" and "lik") are often aligned in the embedding space.

Finally, each model (and each pre-training checkpoint) is fine-tuned for a single epoch on the training set using a cross-entropy loss function computed exclusively on the binary prediction tokens (*yes* or *no*). Further details on training setup and hyperparameters can be found in Appendix A.

3.4 Evaluation

The evaluation is conducted at multiple levels of granularity by computing standard metrics such as accuracy and macro F1-score. Specifically, we measure performance for each position (start, middle, end) and for each n-gram length at each position; furthermore, we conduct these evaluations separately for (1) morphologically simple words ("Non-derived"), for example: "Is *_dr* in *drink*?"; (2) derived words where the asked n-gram does not overlap by length and position to the word's morphemes (abbreviated as "Derived N.M.N.", Non-Morphemic N-gram), for example: "Is *sl* in *dislike*?"; (3) derived words where both position and length of the asked n-gram match exactly the morpheme in the word (abbreviated as "Derived M.N.", Morphemic N-gram), for example: "Is *_dis* in *dislike*?". This separation allows us to assess if PLMs recognize morphemes better or worse than a meaningless substring, with a further linguistic distinction between prefixes, suffixes and roots.

4 Results

Before presenting our experimental results, we note that all models – except for Pythia 70M – achieved near-perfect accuracy ($\geq 99\%$) on the identity set (44 110 examples). Since these cases rely solely on identity matching rather than character-level knowledge, the following sections focus exclusively on non-identity cases.

4.1 Results on the fully pre-trained models

Table 1 presents the overall F1-score for each model distinguished by morphological category. **Results increase consistently at higher parameter size**, with Pythia-1B obtaining by far the best results. The improvement is especially relevant when comparing 160M, 410M, and 1B to the smallest model, Pythia-70M, which lags behind by a substantial gap. **Across all models, F1-score is higher for morphemic n-grams of characters** rather than meaningless substrings, especially in the derived words class.

Table 2 provides a fine-grained overview of performance across position, substring length and morphological value[4]. Position considerably influences model performance: **substrings at the beginning of a word are recognized the most**, with Pyhtia-1B and 410M achieving F1 $> 90\%$, followed by the end of word n-grams as the second best performing position. Pythia-70M, on the other hand, deviates from this trend, performing better on end-position substrings than on start-position ones. **Substrings in middle position are the hardest to identify**, for which each model obtained a substantially lower F1. The length of the n-gram also affects performance: **longer n-grams are generally recognized better than shorter ones**, since they probably carry some semantic signal that facilitates prediction. **Models perform worst on single characters, especially if they are located in middle and end position**. Even the best-performing model, Pythia-1B, achieves only 77% and 79% F1 for these classes, hence a 20% drop from start position n-grams of length 1. This suggests that, as parameter size increases, **character-blind PLMs exhibit a high awareness of the starting letter of a word but significantly lower awareness of single characters at middle and end positions**[5].

Importantly, measuring F1 only on the substrings in start position that begin with {a, h, o, v}, which we excluded from the training set (see Section 3.3), there is no significant drop in performance (max -4% for Pythia-70M), indicating that models generalized the task and are effectively leveraging information acquired during pre-training (see Appendix C).

Looking at the morphological classes, the evident increase for morphemic n-grams (underlined

[4]For more details on statistics about each class described in Table 2, see Appendix B.

[5]See Figure 7 in Appendix for results on single characters.

		Morph. simple				Derived N.M.N.				Derived M.N.			
	len	**70M**	**160M**	**410M**	**1B**	**70M**	**160M**	**410M**	**1B**	**70M**	**160M**	**410M**	**1B**
start	**1**	0.63	0.82	0.95	0.97	0.59	0.76	0.92	0.95	0.54	0.75	0.91	0.93
	2	0.52	0.84	0.92	0.96	0.51	0.81	0.88	0.93	0.57	0.9	0.93	0.94
	3	0.59	0.88	0.94	0.96	0.57	0.86	0.92	0.95	0.58	0.89	0.95	0.96
	4	0.62	0.91	0.96	0.97	0.6	0.89	0.94	0.97	0.58	0.91	0.96	0.98
	5	0.63	0.93	0.96	0.98	0.61	0.91	0.95	0.97	0.56	0.91	0.97	0.97
	6	0.63	0.94	0.97	0.98	0.61	0.92	0.96	0.98	/	/	/	/
	all	0.62	0.88	0.95	0.97	0.58	0.86	0.93	0.96	0.57	0.89	0.94	0.95
middle	**1**	0.58	0.68	0.72	0.77	0.59	0.66	0.71	0.76	/	/	/	/
	2	0.6	0.7	0.7	0.77	0.6	0.67	0.65	0.74	/	/	/	/
	3	0.58	0.73	0.71	0.79	0.59	0.68	0.67	0.75	0.60	0.82	0.79	0.91
	4	0.57	0.75	0.75	0.83	0.57	0.71	0.71	0.78	0.63	0.85	0.84	0.91
	5	0.56	0.77	0.76	0.85	0.57	0.72	0.74	0.81	0.66	0.87	0.86	0.92
	6	0.55	0.8	0.79	0.87	0.57	0.75	0.77	0.84	0.62	0.90	0.88	0.92
	all	0.58	0.72	0.72	0.79	0.59	0.69	0.7	0.76	0.64	0.86	0.85	0.91
end	**1**	0.67	0.77	0.71	0.79	0.62	0.68	0.65	0.73	0.67	0.75	0.68	0.78
	2	0.71	0.8	0.75	0.83	0.65	0.75	0.69	0.81	0.82	0.90	0.83	0.91
	3	0.74	0.83	0.81	0.88	0.73	0.81	0.76	0.86	0.86	0.89	0.78	0.89
	4	0.78	0.87	0.87	0.92	0.8	0.88	0.84	0.91	0.86	0.91	0.87	0.94
	5	0.82	0.9	0.9	0.94	0.82	0.89	0.89	0.93	/	/	/	/
	6	0.84	0.91	0.93	0.96	0.84	0.9	0.9	0.94	/	/	/	/
	all	0.75	0.83	0.8	0.87	0.74	0.82	0.79	0.86	0.82	0.86	0.8	0.9

Table 2: Test set F1-score obtained by each Pythia model across different positions, lengths and morphological categories. Some cells in the derived (morphemic n-gram) column reports "/" since there are not enough morphemes, specifically if less than 100, for such category. We underline scores for morphemic n-gram of characters that obtained higher results than meaningless substrings. Derived N.M.N. are non-morphemic n-grams, while Derived M.N. are morphemic n-grams: for explanation, see sect. 3.4.

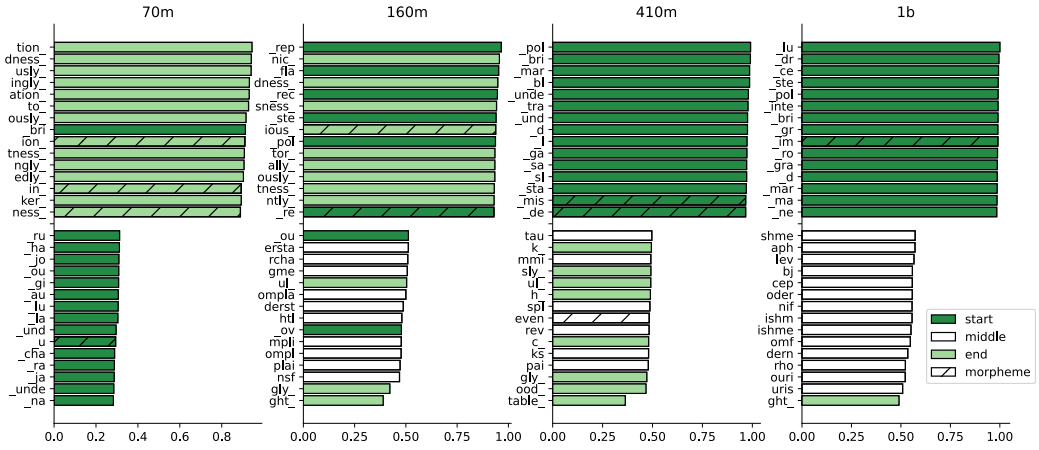

Figure 1: Top/bottom 15 n-grams for each model distinguishing by position (color) and by whether it might have morphological value (dashing). Each n-gram reported has a minimum of 100 occurrences (50 positives and 50 negatives).

in Table 2) varies with respect to position and length: the gain on prefixes is only marginal, but more marked for smaller models and for prefixes of length 2, which are the most productive (*un-*, *im-*, *re-*, etc.). On the other hand, **performance on roots, which carry strong semantic information, is substantially higher than meaningless middle-position substrings**. Similarly, **suffixes, which are closely related to morphosyntax** (*-ly*, *-er*, *-ness*), **are recognized more accurately than meaningless end-position substrings**. Interestingly, performance degrades on derived words when the queried n-gram is not a morpheme ("Derived N.M.N." in Table 2), suggesting that, for morphologically complex words, models are more aware of morphemes than of meaningless substrings.

Although bigger models perform better, Pythia-410M, curiously, deviates from the size-performance rank only on suffixes, performing worse than the 160M and 70M models. One hypothesis for this phenomenon could be that Pythia-410M has a 2x deeper architecture (24 layers) than the 160M model, leading to a different convergence dynamic.

4.1.1 Focus on character n-grams

A more in-depth analysis confirms the relation between parameter size and n-gram position. Figure

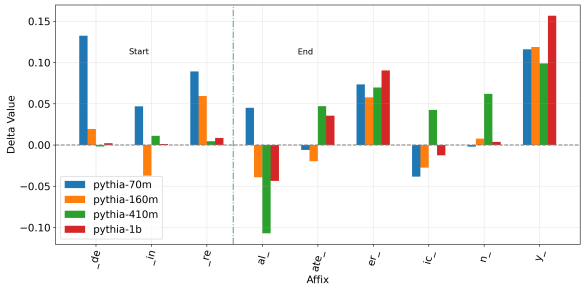

Figure 2: Difference in F1 score performance on the n-gram when it has morphemic value and whenever it occurs without a morphological function. Positive values indicate a performance gain when the n-gram is morphemic.

1, which represents the 15 best and worst performing n-grams for each model, shows that as model size increases, the top 15 shift from mostly end-position n-grams (light green bar) to mostly start-position n-grams (dark green bar). We can also see that the bottom class is mostly populated by middle n-grams (white bar), except for Pythia-70M. It is likely that smaller models perform better on end n-grams as they might be leveraging morphosyntactic information to recover suffixes. This is shown by the fact that, although the best-performing n-grams are not often morphemes (marked with hatchings), many n-grams in the top 15 are either substrings or superstrings of affixes, sometimes even combining two degrees of derivation. For example, in the 70M and 160M models several n-grams include *ly* (*-tly*, *-ingly*, *-ally*, *-edly*, etc.), while in the 410M and 1B models some top performing n-grams correspond to the beginning of prefixes (*und-*, *unde-*, *inte-*). **Despite not necessarily segmenting words precisely at linguistic boundaries, it seems that morphemic information is nonetheless valuable for the models. This is especially true for smaller ones**, which perform better on suffixes, carrying important morphosyntactic information, while bigger models gain awareness of prefixes, which have a more semantic impact on the word.

It is also worth noting that, when n-grams have both a morphemic and non-morphemic function (fast-*er* vs. corn-*er*), there are some performance differences. The extent of this delta is, however, very affix and model specific. We report in Figure 2 the normalized delta in F1 score for such cases[6]. We selected only n-grams of characters that have at least 100 positive and 100 negative examples, both when used as a morpheme and when used as a meaningless substring (therefore, a minimum of 200 vs 200 examples). The n-grams *-er* and *-y*, when acting as suffixes, increase the F1 on all models tested, suggesting a strong sensitivity to the morphosyntactic role of these affixes. On the other hand, among the observed prefixes, performance gain in F1 is generally restricted to the 70M model. Due to the small number of considered n-grams, further research is required to extend this analysis and effectively assess the impact of morphological value for homographic n-grams.

[6]The delta is normalized against the maximum F1 between morphemic and non morphemic n-gram score.

4.2 On the emergence of character-level competence

Figure 3 presents the performance of each pre-trained checkpoint fine-tuned on our task, distinguishing by position and morphological value. Overall, the earliest pre-training steps – including random initialization – perform at or below a majority class baseline, with no meaningful fluctuations. Each model begins to show an increase in F1-score between steps 32 and 512, corresponding to approximately 64 million to 1 billion tokens of pre-training. These results confirm that our fine-tuning approach does not inject enough character-related knowledge to solve the task, hence providing strong evidence that **pre-training with a language modeling objective implicitly induces some character-level knowledge**, although a **substantial amount of data is required for the emergence of this competence and significantly more is needed to reach higher performance**, showing the emergent nature of this ability in character-blind PLMs. **Model size does not influence the point of emergence, but bigger models achieve higher results**, although with no substantial performance gap between the 160M, 410M, and 1B models. The trends in which this competence emerges vary based on position and morphological value: **performance generally improves earlier for morphemes** (dotted lines in Figure 3) **than for meaningless substrings, particularly for suffixes and roots**. Notably, performance for start and middle positions continues to improve beyond step 3000, except for Pythia-70M which achieves significantly lower scores and reaches an early performance plateau on these positions. In contrast, **awareness on suffixes, which are closely related to morphosyntax, emerges much earlier**, around steps 512 and 1000 reaching a plateau between steps 2000 and 3000. Interestingly, after step 3000, even Pythia-70M ex-

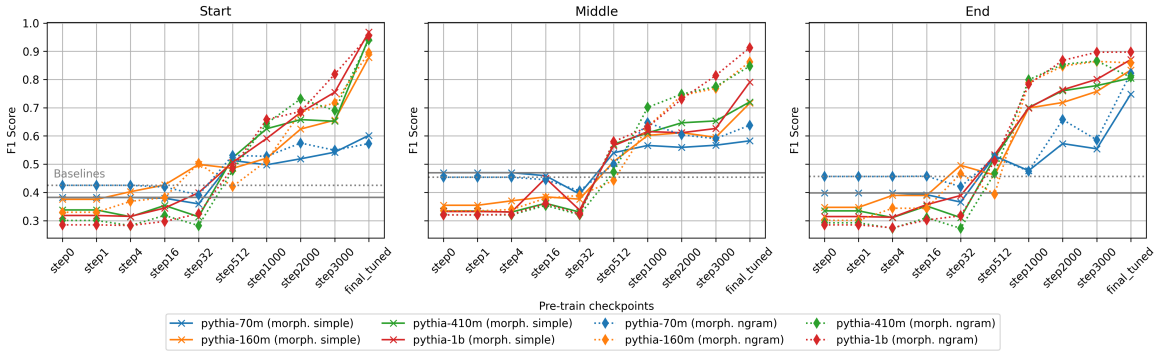

Figure 3: F1-scores obtained by each pre-train checkpoint on our test set. Position is distinguished column-wise, linestyles and markers distinguish between n-grams of morphologically simple (non-derived) words and morphemic n-grams of derived words. Baselines reported for this distinction are obtained by predicting always "yes".

hibits a sharp increase in F1-score only for the end position, particularly on suffixes, suggesting that even small models can acquire an understanding of suffixes given enough pre-training data. In middle position, **performance on roots, which carry strong semantic information, is substantially higher and emerges earlier than meaningless substrings**, on which models struggle the most. After step 3000, the steep increase in performance of bigger models for the start and middle position of non-morphemic n-grams suggests that the amount of pre-training data is essential for developing character awareness, especially for non linguistically relevant substrings. Pythia-1B improves significantly on non-morphemic n-grams in middle position, by far the most difficult class across all models. This suggests that **awareness of n-grams in middle position, not so relevant to solve language modeling, benefits not only from the amount of pre-training data but also from model size**.

4.2.1 Focus on character n-grams

To have a closer look at the emergence of model's competence during pre-training according to the different n-gram positions considered, we report in Figure 4 the percentage of each position among the best 100 performing n-grams. After step 512, before which models perform below baseline, a peculiar trend emerges for each size: while middle position n-grams get less represented among the top 100, end position substrings start to emerge in the immediately following checkpoints (red line in Figure 4). But, for bigger models, after step 3000, start positions n-grams become the most represented (blue line). Specifically, at the end of pre-training, the best 100 performing n-grams

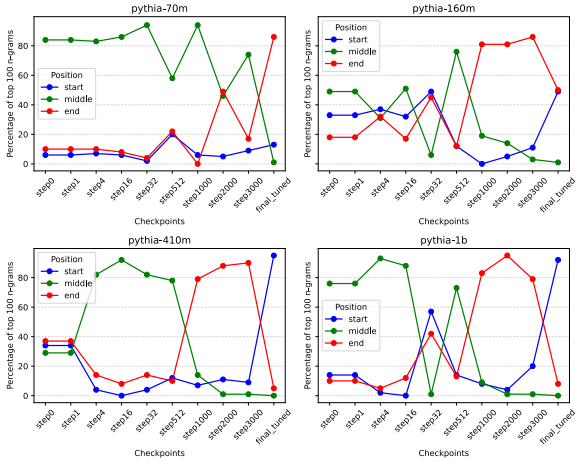

Figure 4: Percentage of top-100 n-grams (start, middle and end) for each model and pre-training checkpoint.

for the Pythia-70M are prevalently in end position, Pythia-160M reaches a balanced presence between start and end position substrings, while for both Pythia-410M and 1B the top 100 gets almost entirely populated by n-grams in start position. This trend suggests a strong relation between model size, n-gram position, and the emergence of character-level awareness: **the character composition at the end of a word is learned earlier during pre-training and small models perform best in such class. Later, as the pre-training data increases, bigger models perform better on substrings located at the beginning of a word.**

4.3 Impact of linguistic features

Our results highlight a clear role of morphological information in substring recognition. For this reason, we further investigated whether other linguistic parameters also influence performance. To this end, we selected several linguistic features and

		n-gram length	word length	word freq	Freq HAL	FamSize	PFMF	P	P*
morph. simple	70M	0,04	-0,02	-0,04	-0,01	-0,01	0,00	/	/
	160M	0,16	-0,04	-0,08	-0,09	-0,05	0,00	/	/
	410M	0,14	-0,08	-0,07	-0,08	-0,04	0,00	/	/
	1B	0,17	-0,08	-0,09	-0,09	-0,04	0,00	/	/
derived (non-morphemic n-gram)	70M	0,03	-0,03	-0,03	0,05	0,06	n.s.	-0,02	0,08
	160M	0,14	-0,07	-0,05	0,04	0,07	0,01	-0,01	0,01
	410M	0,12	-0,09	-0,05	0,04	0,07	0,01	0,02	-0,09
	1B	0,14	-0,09	-0,08	0,03	0,07	0,03	0,04	-0,07
derived (morphemic n-gram)	70M	0,02	n.s.	-0,08	0,14	0,18	0,04	n.s.	0,13
	160M	0,10	n.s.	-0,14	-0,06	-0,04	0,09	0,06	-0,06
	410M	0,12	-0,02	-0,18	-0,19	-0,13	0,12	0,15	-0,14
	1B	0,14	n.s.	-0,22	-0,10	-0,06	0,16	0,08	-0,07

Table 2: Effect size, measured through rank biserial correlation, of variation between the class of correct predictions and the class of incorrect predictions. Backslash indicates that the feature does not apply to that class, while "n.s." indicates that variation (measured through the Mann-Whitney U-test) was not statistically significant.

examined whether their presence differs significantly between correctly and incorrectly predicted instances. To assess statistical significance, we applied the Mann-Whitney U-test to compare the two groups. When significant differences were found, we computed the rank biserial correlation to measure effect size. The selected features include n-gram length, word length, and word frequency, using Wikipedia[7] as the reference corpus. Additionally, we incorporated morphology-based features derived from MorphoLex, which computes frequency values based on the English Lexicon Project (ELP). Among these features, we considered the frequency ranking of a morpheme within its morphological family (PFMF), the total number of words sharing the same morpheme (FamSize), and the summed token frequency of all words containing the morpheme (FreqHAL). We also included two measures of morphological productivity: the likelihood that a word containing a given morpheme is a neologism (P) and the likelihood that a neologism contains a given morpheme (P*).

The results of our analysis are presented in Table 2. Variation is almost always statistically significant, and most exceptions concern word length when testing derived words with morphologically motivated n-grams. These findings further confirm the influence of linguistic features on models' ability to recover character-level information. The lack of significance in these specific cases suggests that **word length, which is typically a predictive measure, becomes less relevant when clear linguistic information is present**.

The effect size is strongest for n-gram length, which consistently yields high results across morphological classes for all models except the 70M. The positive direction indicates that longer n-grams are more likely to be correctly predicted. Notably, n-gram length is the only feature where the effect size is lowest on derived words with morphemic n-grams and highest on morphologically simple words, also indicating that clear linguistic information reduces the relevance of length effects.

[7]We used the 2021 English Wikipedia dump.

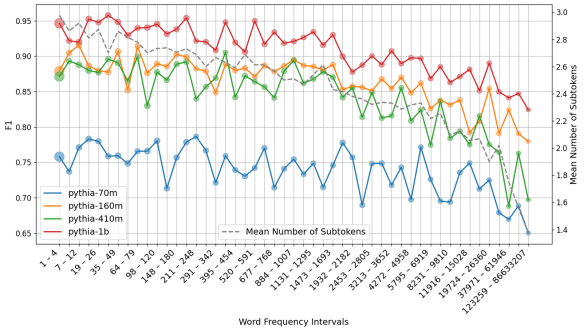

Figure 5: F1-score across frequency intervals for each model on morphemic n-grams. The dashed line indicates the average number of subtokens per frequency bin. The marker size reflects the number of data points, where the first frequency bin has 1198 occurrences and the last one has 254.

Concerning frequency-effects variables, effect sizes are more pronounced for morphemic n-grams. In particular, for this class, high PFMF[8] and low word frequency impact performance, with the effect strengthening as model size increases. This suggests that **lower-frequency words and morphemes facilitate the identification of morphemic substrings**. A negative correlation between substring identification and word frequency, without morphological distinction, was also found by Kaushal and Mahowald (2022).

Figure 5 shows the trend between word frequency and performance of morphemic n-grams, reporting also the average number of subtokens

[8]The PFMF feature, whose effect on performance has a positive direction (opposite to word frequency), is expressed as a ranking, with lower values indicating higher frequency.

per frequency interval[9] (represented by the dashed line): single-subtoken words perform worst, while multi-subtoken words perform best. This observation is supported by a strong positive Spearman correlation across all models between the average F1-score and the number of subtokens per frequency bin[10] (for the 70M, 160M, 410M, 1B models, respectively, $\rho = .63$, $\rho = .77$, $\rho = .82$, $\rho = .85$, all $p < 0.005$). These results further suggest that **models access morphological information in low-frequency words more easily when they are segmented into multiple subtokens**. This effect is more accentuated in bigger models.

5 Conclusion

Our study explores whether and how character-blind PLMs acquire knowledge of the character composition of words, a competence often dubbed the *Spelling Miracle*. By systematically evaluating a family of decoder-only Transformers on a simple yet controlled binary substring identification task, we shed light on the extent to which these models internalize character-level structure, how this phenomenon evolves during pre-training, and the influence of linguistic features such as morphological composition and frequency.

Our findings confirm that substantial amounts of data are necessary for character knowledge to emerge. Although the point of initial emergence is roughly consistent across parameter scales, larger models exhibit superior final performance, underscoring the interaction between data scale and model size. Moreover, the trend of emergence varies depending on n-gram position, morphological value, and model size. Morphemic substrings, especially suffixes and roots, carrying morphosyntactic and semantic information, are recognized earlier and better compared to meaningless substrings in the same positions. On the other hand, bigger models develop a high awareness of the starting n-grams of a word as pre-training data increases. Performance advantages for morphemic n-grams are especially evident in low-frequency words, which often split into multiple subtokens, suggesting that segmentation may guide models to develop more character-informed representations. These results highlight the emergent nature of character-level awareness in character-blind PLMs. Methodologically, our experiments suggest that even a simple fine-tuning approach can serve as a diagnostic tool for knowledge acquired during pre-training, offering a complementary approach to few-shot prompting and probe-based methods. By systematically dissecting where, when, and how PLMs learn about characters and substrings, our work provides both conceptual and empirical grounding for the *Spelling Miracle*[11].

Limitations and Future Work

While our study provides valuable insights into the ability of pre-trained language models (PLMs) to recognize characters, substrings, and morphemes, it has some limitations that should be acknowledged. First, our experiments focus primarily on substring awareness in PLMs, but we do not explicitly assess how this competence translates into downstream tasks. Future work could explore whether substring sensitivity has a measurable impact on real-world NLP applications.

Another limitation concerns the impact of tokenization strategies. Many PLMs utilize subword tokenization methods like Byte-Pair Encoding (BPE) or WordPiece, which can introduce biases in how substrings are recognized. Investigating this issue with character-level models or alternative tokenization approaches in a follow-up study could help clarify these effects.

Additionally, while we examine when substring awareness emerges during pre-training, our findings are constrained by the availability of specific models and checkpoints. Different architectures, training objectives, or corpora may lead to variations in this phenomenon, requiring broader validation across a wider range of models. Similarly, our analysis is conducted in controlled settings with predefined substrings and morphemes, which may not fully capture the complexity of morphological variation across languages. Extending this work to multiple linguistic families would help assess the generalizability of our conclusions.

Despite these limitations, our findings contribute to a deeper understanding of how PLMs process substrings and morphemes, offering a foundation for future research on character-level linguistic competence in neural language models.

[9] Subtokens are obtained using the Pythia tokenizer.

[10] Bins are computed using quantiles (q = 50) over log-transformed frequency values.

[11] Code is available at the following repository: https://github.com/snizio/Beyond-Spelling-Miracle

Acknowledgments

The authors acknowledge the support of the project XAI-CARE-PNRR-MAD-2022-12376692 under the NRRP MUR program funded by the NextGenerationEU, the LuCET - LingUistic Complexity Evaluation in educaTion - project under the PRIN grant no. 2022KPNY3B funded by the Italian Ministry of University and Research and the PNRR MUR project PE0000013-FAIR. Partial support was also received by the project "Advancing Italian Language Processing with Small-Scale Training and Preference Modeling" (IsCb8_AILP), funded by CINECA under the ISCRA initiative, for the availability of HPC resources and support.

References

Paola Angelelli, Chiara Valeria Marinelli, and Cristina Burani. 2014. The effect of morphology on spelling and reading accuracy: a study on italian children. *Frontiers in Psychology*, 8.

David A Balota, Melvin J Yap, Keith A Hutchison, Michael J Cortese, Brett Kessler, Bjorn Loftis, James H Neely, Douglas L Nelson, Greg B Simpson, and Rebecca Treiman. 2007. The english lexicon project. *Behavior research methods*, 39:445–459.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. In *Journal of machine learning research*.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Lukas Edman, Helmut Schmid, and Alexander Fraser. 2024. CUTE: Measuring LLMs' understanding of their tokens. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3017–3026, Miami, Florida, USA. Association for Computational Linguistics.

Avia Efrat, Or Honovich, and Omer Levy. 2023. LMentry: A language model benchmark of elementary language tasks. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10476–10501, Toronto, Canada. Association for Computational Linguistics.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Jennifer Hu and Roger Levy. 2023. Prompting is not a substitute for probability measurements in large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5040–5060, Singapore. Association for Computational Linguistics.

Jing Huang, Zhengxuan Wu, Kyle Mahowald, and Christopher Potts. 2023. Inducing character-level structure in subword-based language models with type-level interchange intervention training. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12163–12180, Toronto, Canada. Association for Computational Linguistics.

Itay Itzhak and Omer Levy. 2022. Models in a spelling bee: Language models implicitly learn the character composition of tokens. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5061–5068, Seattle, United States. Association for Computational Linguistics.

Ayush Kaushal and Kyle Mahowald. 2022. What do tokens know about their characters and how do they know it? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2487–2507, Seattle, United States. Association for Computational Linguistics.

Md Tahmid Rahman Laskar, Sawsan Alqahtani, M Saiful Bari, Mizanur Rahman, Mohammad Abdullah Matin Khan, Haidar Khan, Israt Jahan, Amran Bhuiyan, Chee Wei Tan, Md Rizwan Parvez, Enamul Hoque, Shafiq Joty, and Jimmy Huang. 2024. A systematic survey and critical review on evaluating large language models: Challenges, limitations, and recommendations. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13785–13816, Miami, Florida, USA. Association for Computational Linguistics.

Paul Lerner and François Yvon. 2025. Unlike "likely", "unlike" is unlikely: BPE-based segmentation hurts morphological derivations in LLMs. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 5181–5190, Abu Dhabi, UAE. Association for Computational Linguistics.

Rosanne Liu, Dan Garrette, Chitwan Saharia, William Chan, Adam Roberts, Sharan Narang, Irina Blok, Rj Mical, Mohammad Norouzi, and Noah Constant. 2023. Character-aware models improve visual text rendering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16270–16297, Toronto, Canada. Association for Computational Linguistics.

Claudia H Sánchez-Gutiérrez, Hugo Mailhot, S Hélène Deacon, and Maximiliano A Wilson. 2018. Morpholex: A derivational morphological database for 70,000 english words. *Behavior research methods*, 50:1568–1580.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Ashima Suvarna, Harshita Khandelwal, and Nanyun Peng. 2024. PhonologyBench: Evaluating phonological skills of large language models. In *Proceedings of the 1st Workshop on Towards Knowledgeable Language Models (KnowLLM 2024)*, pages 1–14, Bangkok, Thailand. Association for Computational Linguistics.

Jonathan Zheng, Alan Ritter, and Wei Xu. 2024. NEOBENCH: Evaluating robustness of large language models with neologisms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13885–13906, Bangkok, Thailand. Association for Computational Linguistics.

A Training Details

The experiments were carried out using two NVIDIA GeForce RTX 4090 GPUs. All models share always the same hyperparameters, except for the learning rate, and tuning strategies. We fine-tuned each model using a linearly decaying learning rate starting from two orders of magnitude lower than the one used during the pretraining of the Pythia models, as follows:

- Pythia-70M: lr = 1e-05
- Pythia-160M: lr = 6e-06
- Pythia-410M: lr = 3e-06
- Pythia-1B: lr = 3e-06

All models were fine-tuned for one epoch using a batch size equal to 16 and the Adam optimizer with no weight decay. The loss function, CE, is computed only on the generated tokens (*yes*, *no*) and not on the entire sequence as usually done with decoder language models. Figure 6 reports validation loss and the number of examples for each validation step.

Importantly, in order to avoid the introduction of the special character "Ġ" to the n-gram, which would signal a token after a whitespace, we add a new special token "[N_GRAM]", whose embedding is trained during fine-tuning, and wrap the queried n-gram like so: "Is [N_GRAM]i[N_GRAM] inside skii?".

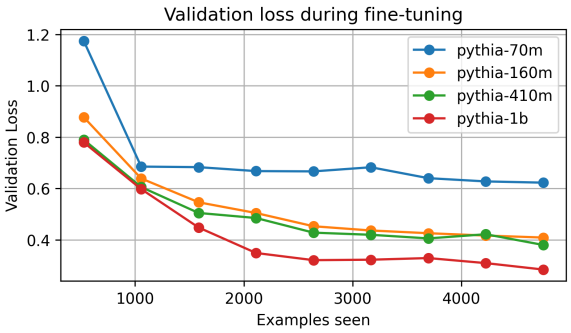

Figure 6: Validation set loss during fine-tuning. The x axis reports the total number of examples for each validation step.

		simple	derived N.M.N.	derived M.N.
start	1	25919	17941	271
	2	25182	16022	2315
	3	23416	16029	1032
	4	20796	16772	590
	5	16007	16296	443
	6	10347	14425	15
	all	121667	97485	4666
middle	1	104492	101835	0
	2	86693	99393	11
	3	60515	81092	260
	4	39222	62927	862
	5	22576	45612	660
	6	11881	30763	373
	all	325379	421622	2166
end	1	27352	17515	1323
	2	25262	14380	5405
	3	22805	13952	3472
	4	19892	14959	2065
	5	15025	15967	54
	6	9837	14198	14
	all	120173	90971	12333

Table 2: Descriptive table showing the instance count for each class.

B Dataset Statistics

In Table 2 are reported the statistics of the dataset used in our experiments for each class considered (positions, lengths, and morphological categories). As mentioned in the main body of the paper, in our evaluation, we discard classes where there are less than 100 occurrences. Specifically, from the MorphoLex dataset we kept words with a PRS (prefix-root-suffix) structure of {0-1-0}, {1-1-0}, {1-1-1}, and {0-1-1}. The number of examples for the motivated n-grams in middle position is low due to the fact that roots in middle position are only taken from the relatively small {1-1-1} subset, which contains words with a prefix, a root, and a suffix. We

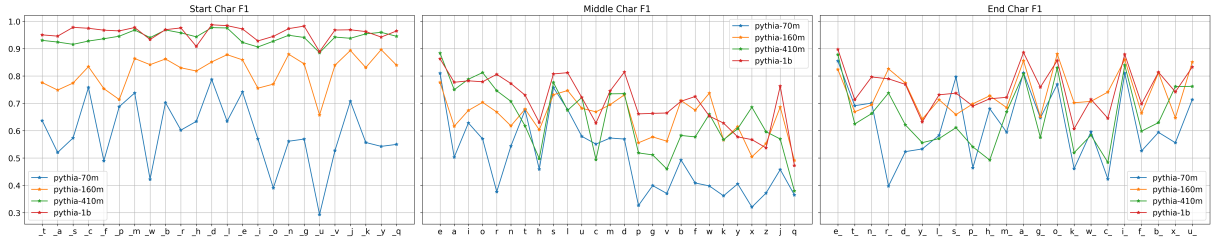

Figure 7: F1 score for each character at each position, sorted by descending character frequency.

excluded from our derived M.N. set roots in start- and end-position: this allowed us to have a perfect comparison strategy between non-morphemic n-grams in start position and prefixes, middle position and roots, end position and suffixes.

C Further details on n-gram results

Figure 7 reports F1 scores for each character, sorted by frequency, at different positions (start, middle, or end of a word), also showing how there is practically no drop on ablated characters {a, h, o, v}. Curiously, a positive trend between character frequency and F1-score is observable only for the middle position.