Scaling Law of Factual Knowledge Injection for LLMs: A Case Study on Arabic Domain

Anonymous ACL submission

Abstract

The growing need for domain-specific large language models (LLMs), underscores the importance of Domain Adaptive Pre-training (DAP) in enhancing downstream task performance. While existing research has established scaling laws for corpus mixture optimization, the scaling laws governing factual knowledge injection remain unexplored. This paper bridges this gap by conducting a case study on Arabic domain-specific factual knowledge injection via DAP. Unlike traditional scaling laws, which rely on token counts and cross-entropy loss, our approach introduces two key innovations: (1) scaling training data based on domain knowledge volume rather than corpus size, and (2) using a knowledge-oriented evaluation method. We developed a scalable data synthesis pipeline that extracts factual knowledge triples from Arabic Wikipedia, generates diverse templates, and populates them to create training data. Experiments on pre-trained models of varying sizes yielded a log-linear scaling trend incorporating model size, knowledge volume, and exposure frequency, indicating a potential practical value in guiding knowledge injection trainings.

1 Introduction

With the rapid development of large language model (LLM) technologies and applications, the demand for domain-specific models continues to grow. Leading domain models (Shi et al., 2024) typically incorporate Domain Adaptive Pre-training (DAP) during training to enhance their performance on downstream tasks (Huang et al., 2023; Bari et al., 2024; Liang et al., 2024). A key role of DAP in enhancing the effectiveness of subsequent fine-tuning is to infuse the model with missing domain knowledge (Wu et al., 2023; Gururangan et al., 2020).

Previous studies have found that for a given set of factual knowledge, their frequencies of repetition ("exposure") during pre-training is crucial for learning effectiveness (Allen-Zhu and Li, 2023a,b, 2024a), and when exposure reaches a certain threshold (e.g., exposure $= 1,000$) can an LLM effectively memorize these factual knowledge. However, a higher exposure count also implies greater data collection/synthesis costs and increased training overhead. In the DAP scenario, more frequent model updates also raise the risk of catastrophic forgetting of the pre-trained model's existing general knowledge (Luo et al., 2023). Therefore, we need to address the following question: *In the DAP scenario, given a model's specific size and a defined amount of domain knowledge to be acquired, how much training data is required to achieve effective knowledge injection?*

Although previous studies have explored data scaling in the context of DAP, they primarily focused on the optimal mixing ratio of general and domain-specific corpora at the token level (Que et al., 2024; Gu et al., 2024), and their approaches do not address the issue of domain knowledge exposure that concerns us. First, the information density varies significantly across different types and

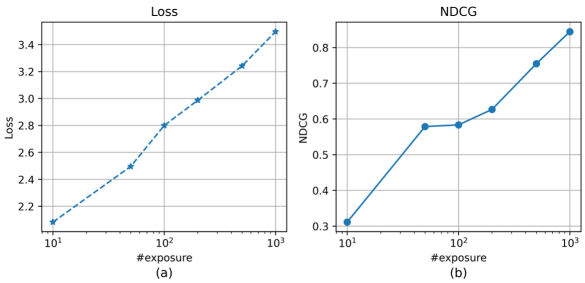

Figure 1: **(a)** Loss values as a function of exposure levels, plotted on a logarithmic x-axis. The y-axis is restricted to the range $[2, 4]$ to highlight variations in loss. **(b)** Normalized Discounted Cumulative Gain (NDCG) scores as a function of exposure levels, also plotted on a logarithmic x-axis. The y-axis is restricted to the range $[0, 1]$, reflecting the typical scale of NDCG values. Both metrics are evaluated across six exposure levels: 10, 50, 100, 200, 500, and $1,000$.

sources of corpora (e.g., factual knowledge density in Wikipedia is much higher than in casual conversation), making it impossible to directly convert the amount of factual knowledge into token counts. Second, since the same factual knowledge can be expressed in multiple ways in natural language, it is challenging to measure the repetition frequency of specific knowledge in the raw corpus. In this paper, we aim to investigate the relationship between the exposure level of knowledge and its injection effectiveness in DAP, using factual knowledge from the Arabic domain as a case study.

To address the limitations of natural corpora, we developed a data synthesis pipeline for generating DAP training corpora for Arabic domain knowledge, enabling precise control over both knowledge quantity and exposure frequency. The pipeline comprises four stages: 1) corpus crawling, 2) extraction of knowledge triples from the corpus inspired by knowledge graph works (Wang et al., 2021; Chen et al., 2024), 3) generation of diverse natural language templates based on the extracted triples by leveraging the approach from (Ge et al., 2024), and 4) synthesis of DAP training corpora with varying exposure levels using the triples and templates. Utilizing this method, we extracted $115,394$ Arabic knowledge triples and synthesized data with up to $1,000$ exposures to conduct knowledge injection experiments across models of varying sizes.

Another critical issue in this work is how to measure to what extend factual knowledge has been successfully injected. In related studies, this is typically achieved through the tail entity prediction task (Geva et al., 2023; Jiang et al., 2019; Dai et al., 2021). Specifically, by designing a prompt that includes the head entity and relation (e.g. "Saudi Arabia", "capital city") of a factual knowledge, the model's probability of predicting the correct tail entity ("Riyadh") or the cross-entropy loss is calculated. So We first evaluate the effectiveness of knowledge injection by computing the cross-entropy loss on the tail entity, as shown in Figure 1 (a). Surprisingly, the loss on the tail entity increases as the number of exposures grows which is contrary to (Allen-Zhu and Li, 2024b). To investigate whether there was an issue with our training process, we try to check whether there is a problem in our training process by checking the rank of the ground-truth token. For this purpose, we employed the NDCG[1] (Normalized Discounted Cumulative Gain) metric, which is commonly used to assess the quality of ranking, with higher values indicating better performance. As shown in Figure 1 (b), our evaluation results reveal that NDCG increases as the amount of exposure increases, suggesting that the training process is functioning correctly. This indicates that higher exposure improves knowledge retrieval effectiveness but leads to a decline in prediction probabilities. We also observed that, although NDCG is a nonlinear metric, it exhibits a clear linear relationship with the exposure level before entering the saturation zone. Our experiments demonstrate that this linear relationship holds across models of different sizes and varying amounts of factual knowledge. Based on these findings, we can use the linear parameters fitted from data with lower exposure levels to predict the required exposure for a target NDCG, thereby guiding the data synthesis and training for knowledge injection in DAP.

Our contributions can be summarized as follows:

- Analyzed the impact of exposure times, model sizes, and knowledge scales on factual knowledge injection during the DAP stage, yielding the following findings: a. Both ranking performance and cross-entropy loss demonstrate a log-linear relationship with exposure times. b. Ranking performance shows a negative log-linear correlation with knowledge sizes and a positive log-linear correlation with model sizes.

- Designed a DAP data synthesis pipeline capable of controlling both the quantity of factual knowledge and the exposure times.

2 Preliminary and Background

2.1 Factual Knowledge

Factual knowledge refers to the collection of objective, verifiable information about the world that is often expressed in structured or semi-structured forms. This type of knowledge encompasses entities, relationships, attributes, and events that can be explicitly stated and retrieved. For instance, factual knowledge can be represented in the form of triples, such as (head, relation, tail). A concrete example of such a triple is (Saudi Arabia, capital

[1]NDCG is a metric used for evaluating the quality of ranking, with values ranging from 0 to 1. An NDCG value of 1, 0.5, and 0.25 corresponds to the rank of the ground-truth token being 1, 3, and 15, respectively. The specific formula can be found in Section 2.3.

city, Riyadh), which captures the factual statement that "the capital of Saudi Arabia is Riyadh."

Such knowledge is typically stored in databases, encyclopedias, or knowledge graphs, making it a foundational element for various applications in natural language processing (NLP), information retrieval, and artificial intelligence (AI). Understanding and leveraging factual knowledge is crucial for tasks like question answering, fact-checking, and semantic search, where accuracy and reliability are paramount.

2.2 Data Synthesis with Personas

To synthesize high-quality and diverse sentence templates for training purposes, we leverage Persona Hub (Ge et al., 2024), a large-scale repository of one billion personas automatically curated from web data. Each persona in the hub represents a unique perspective or identity, enabling the generation of synthetic data that reflects a wide range of linguistic styles, cultural contexts, and individual viewpoints. Our approach involves designing prompts that incorporate specific personas to guide LLM in generating data tailored to distinct narrative voices.

For instance, when tasked with describing a factual event such as someone's birthday, different personas yield markedly varied expressions. An IT programmer might phrase it as "{name} first logged into the world on {birthday}." while a nature photographer could describe it as "{name} captured their first breath of life on {birthday}.". By systematically selecting personas from Persona Hub and embedding them into carefully crafted prompts, we ensure that each generated data point is not only factually consistent but also stylistically rich and contextually nuanced.

This methodology offers two key advantages. First, the diversity of personas ensures that the synthesized data spans a broad spectrum of linguistic patterns and perspectives, enhancing the robustness of downstream models trained on this data. Second, the use of personas introduces an element of creativity and variability that mimics real-world human expression, thereby improving the naturalness and authenticity of the generated content. Overall, our approach demonstrates how Persona Hub can be effectively utilized to produce synthetic data at scale, bridging the gap between structured factual knowledge and expressive, human-like narratives.

2.3 NDCG

Normalized Discounted Cumulative Gain (NDCG) (Järvelin and Kekäläinen, 2002) is a widely used metric for evaluating ranking system quality. It assesses both the relevance scores and positions of items in a ranked list. The relevance score rel_i of an item at position i is discounted logarithmically to penalize lower-ranked relevant items. The NDCG formula is defined as:

$$NDCG = \frac{DCG}{IDCG}$$

where

$$DCG = \sum_{i=1}^{n} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

and

$$IDCG = \sum_{i=1}^{n} \frac{2^{rel_i^*} - 1}{\log_2(i+1)}$$

Here, rel_i is the relevance score of the item at position i in the ranked list, ${rel_i}^*$ is the relevance score in the ideal ranking, and n is the number of items. NDCG ranges from 0 to 1, with higher values indicating better ranking quality.

In our work, we simplify this by focusing solely on the ground-truth token: we assign $rel = 1$ to the ground-truth token and $rel = 0$ to all others. Thus, the NDCG for a single prediction reduces to:

$$NDCG = \frac{1}{\log_2(1 + rank_{gt})}$$

where $rank_{gt}$ is the position of the ground-truth token in the model's output. This simplified formula directly reflects the model's ability to rank the correct token highly, providing a precise measure of its in performance predicting factual knowledge.

3 Scaling Laws of Knowledge Injection

3.1 Ranking Performance Scales Log-Linearly with Exposure Size

As depicted in Figure 2 (a), increasing exposure enhances NDCG for both 7B and 14B models, reflecting improved knowledge injection performance. This aligns with expectations, as more frequent exposure aids in better memorization of factual knowledge. Specifically, the 7B model demonstrates a near log-linear relationship between NDCG and exposure, with minor deviations at $N = 50$. In contrast, the 14B model shows a rising NDCG trend with increasing exposure but approaches saturation at NDCG $= 0.9$ for

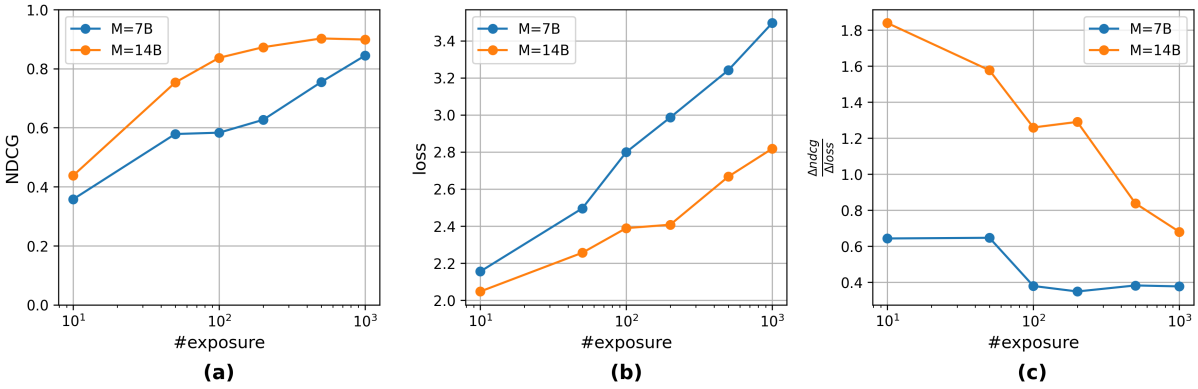

Figure 2: (a) NDCG vs. exposure times; (b) loss vs. exposure times; (c) NDCG and loss increments with regard to the pre-trained model across different exposures. All these curves are obtained with $K = 50,000$. Note: NDCG and loss are computed only on tail entities.

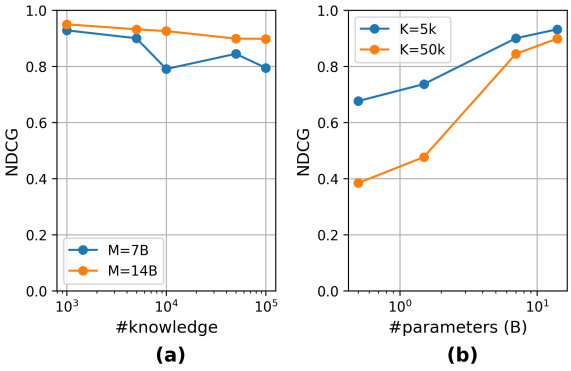

Figure 3: (a) NDCG vs. number of factual knowledge; (b) NDCG vs. number of parameters. All these curves are obtained with $N = 1,000$. Note: NDCG is computed only on tail entities.

$N > 100$. Notably, the 14B model achieves faster NDCG gains with fewer exposures, reaching NDCG $= 0.836$ at $N = 100$, while the 7B model requires $N = 1,000$ to attain a similar level (NDCG $= 0.844$). This suggests the 7B model needs approximately 10 times the exposure to match the 14B model's performance. Given that the 7B model's unit computational cost is roughly half that of the 14B model, its total computational cost to reach this level is about 5 times higher. This proportional relationship is consistent for the 7B model at $N = 500$ and the 14B model at $N = 50$.

3.2 More Exposures Result in Higher Cross-Entropy Loss

In Figure 2 (b), we observe an unexpected increase in loss with rising exposure. However, given the concurrent improvement in NDCG, this loss increase does not signify a decline in knowledge injection effectiveness. For both model sizes, the loss exhibits a near log-linear relationship. Notably, in the $N > 100$ range, while the NDCG growth of the 14B model slows as it approaches its limit, the corresponding loss continues its linear upward trend. Comparing the two models, the 14B model's loss grows at a significantly slower rate than the 7B model. Starting from nearly identical loss values (see Appendix 16 (b)), the 14B model requires 10 times the exposure ($N = 1000$) to reach a loss level comparable to that of the 7B model at $N = 100$. Given that cross-entropy loss naturally reflects model encoding efficiency, this trend suggests that factual knowledge injection imposes a cost on encoding efficiency, with larger models exhibiting a slower degradation in encoding efficiency as exposure increases.

3.3 Cross-Entropy Loss Correlates With Ranking Performance

In Figure 2 (c), we further analyze the trend of the ratio between NDCG and loss increments with regard to the pre-trained model across different exposures. The figure reveals the following insights:

1) For the 7B model, the ratio remains constant within the intervals $N \in [10, 50]$ and $N \in [100, 1000]$, with a decline in the ratio occurring between these intervals, corresponding to the slight deviation from the overall linear trend observed in the NDCG and loss plots at $N = 50$. This phenomenon suggests that the curves in panels (a) and (b) may exhibit piecewise linear characteristics. Specifically, the fluctuations in NDCG and loss between $N = 50$ and $N = 100$, as well as their relative stability outside this range, appear synchronized, indicating a potential deeper connection between the two metrics.

2) For the 14B model, the ratio follows a negative log-linear relationship with increasing exposure. Given the linear trend of the loss, we can infer that the corresponding NDCG trend in this interval can be approximated by a quadratic function. This implies that even for scenarios approaching NDCG saturation, it is possible to predict NDCG results at higher exposures by fitting a quadratic function based on experiments with fewer exposures.

The experimental results for the smaller 1.5B and 0.5B models exhibit similar log-linear scaling trends, albeit with a minimal increase in NDCG and a more pronounced rise in loss, as detailed in Figure 14.

3.4 Ranking Performance Scales Log-Linearly with Knowledge Size

The increase in knowledge volume elevates the learning difficulty, manifesting as a negative log-linear decline in NDCG.

Figure 3 (a) illustrates the relationship between the number of factual knowledge triples and NDCG performance for two model sizes, $M = 7$B and $M = 14$B, with $K = 1,000$. As the number of knowledge triples increases, the NDCG values for both models follow a negative log-linear trend. This indicates that a larger volume of knowledge indeed escalates learning difficulty, with this difficulty growing log-linearly, enabling predictions of large-scale knowledge injection effects based on performance with smaller datasets. Notably, a similar linear trend is absent in the loss, likely because for smaller knowledge volumes, $1,000$ exposures may exceed training needs. As a result, while NDCG saturates, the loss continues to rise. In contrast, for larger knowledge volumes where NDCG remains unsaturated, both loss and NDCG grow simultaneously, preventing a concurrent linear relationship between the two metrics. A similar trend of smaller models of 0.5B and 1.5B can be found in Figure 15 (a).

3.5 Model Capacity Scales Log-Linearly with Parameters

There exists a log-linear relationship between the scale of model parameters and NDCG performance, indicating that the model's knowledge representation capability improves log-linearly with the increase in parameters.

Figure 3 (b) illustrates the relationship between model parameters and NDCG performance across two distinct knowledge scales, $K = 5k$ and $K = 50k$, with $N = 1000$. The curves demonstrate an approximate log-linear relationship between the number of parameters and NDCG scores across both knowledge scales. Similar results can also be observed in for other knowledge scales as illustrated in Figure 15 (b). These findings indicate that the model's effectiveness in capturing and representing factual knowledge often termed its capacity—increases log-linearly with the scale of parameters, rather than adhering to the linear growth typically assumed. Similarly, no concurrent log-linear trend is observed in the loss, likely due to the same underlying reason in knowledge scaling experiments.

4 Domain Knowledge Extraction and Data Synthesis

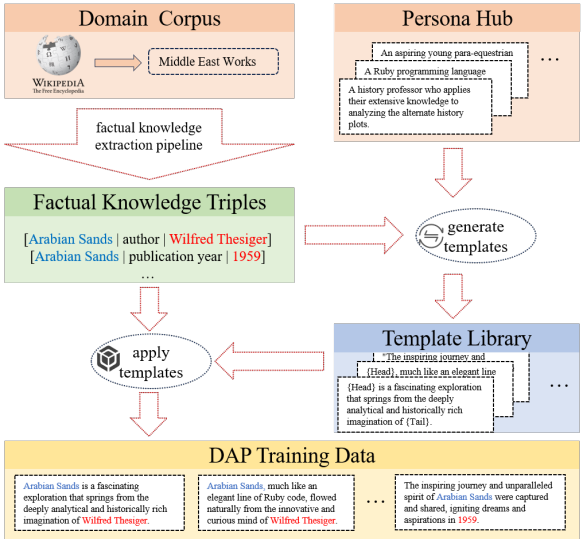

Figure 4: The framework of DAP data synthesis pipeline. Raw text is extracted from Wikipedia and processed through a factual knowledge extraction pipeline to obtain structured triples. These triples are categorized into multiple relation types, which are used to design sentence templates. Using Persona Hub and Qwen2.5-72B-Instruct, tailored prompts are created for each relation type to generate high-quality templates. Finally, the triples are inserted into these templates to produce semantically rich training data.

Conducting scaling law research on domain knowledge injection requires obtaining training data with precise control over both the quantity of knowledge and its exposure frequency. To tackle this challenge, as illustrated in Figure 4, we developed a framework for data synthesis based on domain-specific corpora. This framework consists of two main steps: a) extraction of factual knowl-

edge from the domain, and b) synthesis of training data based on the extracted factual knowledge. Section 4.1 details the multi-stage pipeline for extracting high-quality factual knowledge triples from raw corpora by LLMs. Section 4.2 describes the method for synthesizing training data with precisely controlled exposure levels using these knowledge triples.

4.1 High-Quality Domain Factual Knowledge Extraction

To facilitate scaling law training and evaluation, we define high-quality factual knowledge triplets by the following criteria: 1) The tail entity must be uniquely determinable given the head entity and relation; 2) Both relations and entities should be expressed with clarity and precision; 3) The triplet should contain domain-relevant information. Building upon prior research (Chen et al., 2024) and our empirical observations, we note that LLMs often extract low-quality triplets from open-domain corpora where pre-defined relation scopes are absent. Examples include ("Mike","travels to","New York"), ("brush teeth","time frame","8:00 AM"), and ("Arabic Sands", "is a", "book").

To enhance the quality of triplet extraction, we have developed a multi-stage factual knowledge extraction pipeline. Figure 5 illustrates our four-stage prompting pipeline for extracting and refining these triples from Wikipedia pages. The process begins with Prompt A, which performs initial triples extraction from raw text, generating a comprehensive but potentially noisy set of candidate triples. Recognizing that these initial extractions may contain inconsistencies and inaccuracies, we employ Prompt B to filter out invalid or semantically implausible triples, thereby enhancing data quality. Building upon this filtered set, Prompt C systematically classifies and standardizes relation types, addressing variations in linguistic expression (e.g., "author" vs. "was written by") through manual consolidation into a unified relation schema. Finally, leveraging the refined relation taxonomy, Prompt D re-extracts triples from the original text with improved precision and consistency. This pipeline, progressively refining the extraction process at each stage, ultimately produces a robust dataset for downstream tasks. The complete prompts and implementation details are available in Appendix A.

4.2 Knowledge based Training Data Synthesis

Having obtained the structured triples $\mathcal{T} = \{(h, r, t)\}$, where h and t represent the head and tail entities, and $r \in \mathcal{R}$ denotes the relation type, our objective is to synthesize these triples into natural language training data for DAP, ensuring scalable exposure times for each factual knowledge piece. Previous studies (Allen-Zhu and Li, 2023a; Dubey et al., 2024) have underscored the critical role of knowledge expression diversity in training efficacy, posing a significant challenge: generating large-scale, diverse yet semantically natural expressions for each knowledge. As depicted in Figure 4, to enhance expression diversity, we adopt the approach from (Ge et al., 2024), leveraging the extensive persona descriptions in Persona Hub to create sentence templates for data synthesis. To maintain semantic coherence, these templates are generated separately for each distinct relation.

To transform structured knowledge into linguistically diverse yet semantically consistent text, we construct a template library for each relation type $r_i \in \mathcal{R}$. Specifically, for each relation r_i, we design a prompt that leverages persona descriptions from Persona Hub. These persona data are used to populate the prompts, which are then processed by Qwen2.5-72B-Instruct to generate a template library $\mathcal{T}_{r_i}$ containing $1,000$ unique natural language templates. For example, for the relation "birth year", the library includes templates such as "{name} was born in {year}" and "{name} first logged into the world in {year}." The prompt and generated template examples can be found in 10 and 11.

Once the template libraries are constructed, we proceed to generate the training dataset $\mathcal{D}_{\text{train}}$, which consists of multiple subsets $\mathcal{D}_K^N$ with varying numbers of knowledge points (K) and exposures per point (N). The generation process involves the following steps:

Template Sampling. For each relation r_i, we randomly sample N templates from its corresponding template library $\mathcal{T}_{r_i}$ to form a candidate template set $\mathcal{S}_{r_i}$:

$$\mathcal{S}_{r_i} \subseteq \mathcal{T}_{r_i}, \quad |\mathcal{S}_{r_i}| = N.$$

This ensures that each relation is represented by a diverse subset of templates while maintaining controlled exposure.

Triples Selection and Instantiation. We randomly select K triples from $\mathcal{T}$, ensuring that the

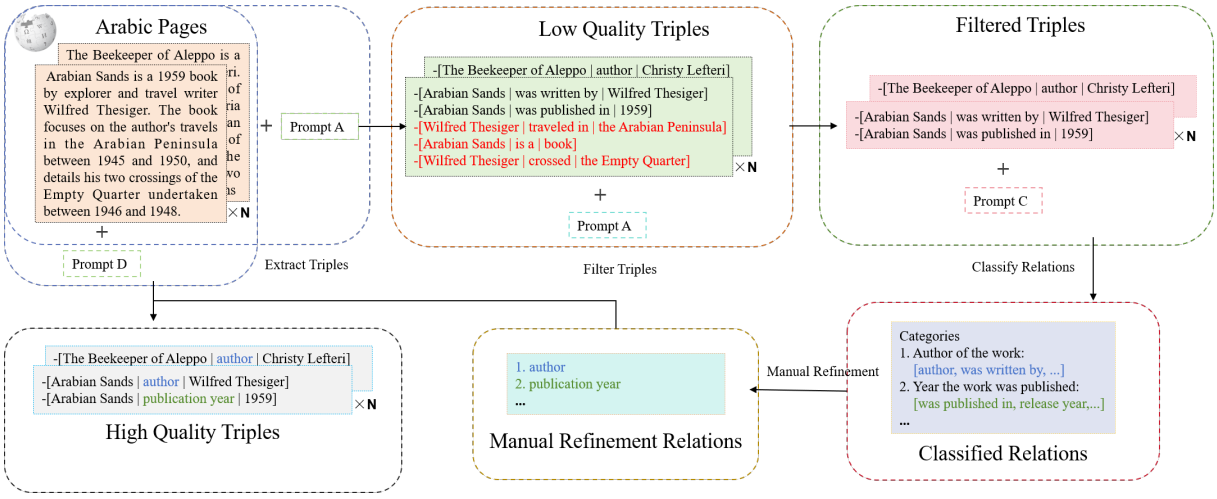

Figure 5: Factual knowledge extraction pipeline. The process begins by extracting low-quality triples from Arabic pages using Prompt A. These triples are filtered using Prompt B to remove invalid triples (red-highlighted examples). The filtered triples are then categorized based on their relations using Prompt C, such as "author" (blue) and "publication year" (green). Manual refinement unifies variations of the same relation within each category. These refined relations are embedded into Prompt D to re-extract high-quality, standardized triples from the original pages, ensuring structured and accurate factual knowledge construction.

selected triples may span multiple relations. Each selected triple is instantiated using all N templates in its corresponding candidate template set $\mathcal{S}_{r_k}$. This process ensures that each knowledge point appears exactly N times in the training data, with each occurrence expressed through a distinct template.

The resulting training dataset $\mathcal{D}_{\text{train}}$ is a union of multiple subsets $\mathcal{D}_K^N$, each corresponding to a specific combination of K knowledge points and N exposures:

$$\mathcal{D}_{\text{train}} = \bigcup_{(K,N)\in\mathcal{C}} \mathcal{D}_K^N,$$

where $\mathcal{C}$ represents the set of all combinations of K and N used in the dataset. The number of relations represented in each subset depends on the random sampling of triples, ensuring diversity in the types of knowledge points included.

This approach balances controlled knowledge injection with linguistic diversity, enabling systematic evaluation of knowledge acquisition during DAP. Additional template examples and implementation details are provided in Appendix A.

5 Knowledge Injection Training and Evaluation

5.1 Knowledge Injection Training

Data Setup. To ensure the quality and efficiency of knowledge extraction, while ensuring the source data can be publicly accessed by the research We have prepared five different scales of factual knowledge triples, which are extracted from Arabic-related pages on Wikipedia. For each scale of triples, we employed six different numbers of templates to augment the data. This process resulted in downstream datasets with varying levels of exposure (N) and different amounts of knowledge (K). (For more details, please refer to Appendix **??**)

Model Setup. For the pre-trained models, we selected the Qwen-2.5 series, which has demonstrated outstanding performance in English-language tasks. Additionally, this series offers a range of open-source pre-trained base models in various sizes. In our experiments, we utilized Qwen-2.5-0.5B, Qwen-2.5-1.5B, Qwen-2.5-7B, and Qwen-2.5-14B as the foundational models for continued pre-training in the downstream domain of Arabic factual knowledge.

DAP Training Setup. We set the learning rate to $7e-6$ for all experiments. For data with different exposure frequencies, we used different global batch size values to ensure sufficient updates during the training process (for details on the specific hyperparameter settings, see Appendix B).

When handling different exposure frequencies (including 10, 50, 100, 200, 500, and $1,000$), we found that directly saving intermediate checkpoints from the training process with an exposure frequency of $1,000$ may not meet the requirements for low exposure counts (such as 10, 50, and 100).

This is because, in some cases, when the exposure frequency is low, it is not possible to precisely obtain the corresponding checkpoints. To address this issue, we conducted separate training sessions for datasets with low exposure counts and only saved the final checkpoint, ensuring that the model achieves optimal performance within the limited number of training steps.

Data concatenation. In our experiment, the average number of tokens per data sample is 32, with the maximum sequence length set to $2,048$. When performing data concatenation, we followed the approach used in DeepSeek-V3 (Liu et al., 2024) to ensure the integrity of the content was preserved.

Computer resources. Our main experiment requires approximately 120 hours of runtime on 8 A100s.

5.2 Knowledge Injection Performance Evaluation

To evaluate the knowledge injection performance, we employ two complementary evaluation metrics: cross-entropy loss and NDCG. These metrics provide insights into both the probabilistic confidence of predictions and the ranking quality of retrieved knowledge.

In this setup, the model is tasked with predicting the tail entity t_i given the head entity h_i and relation r_i, simulating a knowledge retrieval task. To guide the model in understanding the task, we adopt an in-context learning approach by constructing a prompt (detailed in Figure 13) that includes the query triple with the tail entity masked $(h_i, r_i, ?)$ along with several exemplar triples containing correct tail predictions. This design ensures that the model recognizes the need to predict the tail entity while leveraging the provided examples as task demonstrations.

For cross-entropy loss evaluation, we compute the average token-level loss across all tokens in the tail entity. This metric reflects the model's uncertainty in its predictions, providing a probabilistic measure of how well the factual knowledge is encoded or compressed by the model.

For NDCG evaluation, we calculate the rank of each token in the predicted tail within the model's vocabulary and derive the corresponding NDCG score. To ensure robustness, we take the minimum NDCG value across all tokens in the predicted tail as the representative score for the triple. This approach penalizes errors in any part of the tail prediction, ensuring a conservative assessment of retrieval accuracy. Finally, we average the NDCG scores across all triples to obtain an overall measure of the model's knowledge retention.

6 Related Works

DAP Related Scaling Laws. Recent studies have advanced continuous pre-training optimization. (Que et al., 2024) introduces the D-CPT and Cross-Domain D-CPT Laws, reducing training costs while enhancing domain-specific and general performance. Similarly, (Gu et al., 2024) proposes the CMR Scaling Law to balance general and specialized capabilities. In cross-lingual CPT, (Zheng et al., 2024) optimizes resource allocation for new languages, while (Kaplan et al., 2020) highlights that larger models achieve equivalent performance with fewer resources, emphasizing the importance of scale.

Knowledge Injection and Data Synthesis. Factual knowledge acquisition in language models is closely tied to training data diversity (Allen-Zhu and Li, 2023a). Allen-Zhu et al. (Allen-Zhu and Li, 2024b) show each Transformer parameter stores 2 bits of knowledge, linking model size to knowledge capacity. Geiping et al. (Geiping et al., 2022) reveal data augmentation improves generalization through scaling laws and regularization. In synthetic data, Tencent AI Lab (Ge et al., 2024) uses a Persona-driven approach with one billion virtual personas to generate high-quality, diverse data, advancing data diversification techniques.

7 Conclusion

This study systematically investigates scaling laws for factual knowledge injection in Arabic large language models through domain-adaptive pretraining. We develop a novel data synthesis pipeline that enables precise control over knowledge quantity and exposure frequency. Our analysis reveals three fundamental log-linear relationships: (1) higher exposure frequency enhances knowledge retrieval while reducing prediction probabilities; (2) increased knowledge scale leads to logarithmic growth in learning difficulty; (3) larger model size significantly improves fact capture capability. These findings provide both theoretical insights and practical guidance for optimizing knowledge injection in DAP frameworks, particularly for low-resource languages like Arabic.

Limitation

Our study explores factual knowledge injection during DAP and introduces a data synthesis pipeline, but several limitations remain. 1. Focusing on the Arabic domain may limit generalizability, requiring exploration across diverse languages and domains. 2. Smaller models need higher exposure levels, highlighting the need to optimize training for varying model capacities. 3. Frequent updates risk catastrophic forgetting, calling for techniques like regularization or memory replay to balance knowledge retention. Addressing these challenges will improve domain-specific language models.

References

Zeyuan Allen-Zhu and Yuanzhi Li. 2023a. Physics of language models: Part 3.1, knowledge storage and extraction. *ArXiv*, abs/2309.14316.

Zeyuan Allen-Zhu and Yuanzhi Li. 2023b. Physics of language models: Part 3.2, knowledge manipulation. *ArXiv*, abs/2309.14402.

Zeyuan Allen-Zhu and Yuanzhi Li. 2024a. Physics of language models: Part 3.3, knowledge capacity scaling laws. *ArXiv*, abs/2404.05405.

Zeyuan Allen-Zhu and Yuanzhi Li. 2024b. Physics of language models: Part 3.3, knowledge capacity scaling laws. *arXiv preprint arXiv:2404.05405*.

M Saiful Bari, Yazeed Alnumay, Norah A. Alzahrani, Nouf M. Alotaibi, Hisham A. Alyahya, Sultan Alrashed, Faisal A. Mirza, Shaykhah Alsubaie, Hassan A. Alahmed, Ghadah Majid Alabduljabbar, Raghad Alkhathran, Yousef Almushayqih, Raneem Alnajim, Salman Alsubaihi, Maryam Al Mansour, Majed Alrubaian, Ali Alammari, Zaki Alawami, AbdulMohsen O. Al-Thubaity, Ahmed Abdelali, Jeril Kuriakose, Abdalghani Abujabal, Nora Al-Twairesh, Areeb Alowisheq, and Haidar Khan. 2024. Allam: Large language models for arabic and english. *ArXiv*, abs/2407.15390.

Hanzhu Chen, Xu Shen, Qitan Lv, Jie Wang, Xiaoqi Ni, and Jieping Ye. 2024. Sac-kg: Exploiting large language models as skilled automatic constructors for domain knowledge graph. In *Annual Meeting of the Association for Computational Linguistics*.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. 2021. Knowledge neurons in pretrained transformers. *ArXiv*, abs/2104.08696.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony S. Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Bap tiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Cantón Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab A. AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriele Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guanglong Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Laurens Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Ju-Qing Jia, Kalyan Vasuden Alwala, K. Upasani, Kate Plawiak, Keqian Li, Ken-591 neth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuen ley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Babu Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melissa Hall Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri S. Chatterji, Olivier Duchenne, Onur cCelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasić, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Ro main Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Chandra Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whit ney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Gold-

schlag, Yashesh Gaur, Yasmine Babaei, Yiqian Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zhengxu Yan, Zhengxing Chen, Zoe Papakipos, Aaditya K. Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adi Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Ben Leonhardi, Po-Yao (Bernie) Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Shang-Wen Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzm'an, Frank J. Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory G. Sizov, Guangyi Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Han Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kaixing(Kai) Wu, U KamHou, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, A Lavender, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollár, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sung-Bae Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Andrei Poenaru, Vlad T. Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xia Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The llama 3 herd of models. *ArXiv*, abs/2407.21783.

Tao Ge, Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. 2024. Scaling synthetic data creation with 1,000,000,000 personas. *arXiv preprint arXiv:2406.20094*.

Jonas Geiping, Micah Goldblum, Gowthami Somepalli, Ravid Shwartz-Ziv, Tom Goldstein, and Andrew Gordon Wilson. 2022. How much data are augmentations worth? an investigation into scaling laws, invariance, and implicit regularization. *arXiv preprint arXiv:2210.06441*.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. *ArXiv*, abs/2304.14767.

Jiawei Gu, Zacc Yang, Chuanghao Ding, Rui Zhao, and Fei Tan. 2024. Cmr scaling law: Predicting critical mixture ratios for continual pre-training of language models. *ArXiv*, abs/2407.17467.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. *ArXiv*, abs/2004.10964.

Huang Huang, Fei Yu, Jianqing Zhu, Xuening Sun, Hao Cheng, Dingjie Song, Zhihong Chen, Abdulmohsen

Alharthi, Bang An, Ziche Liu, Zhiyi Zhang, Junying Chen, Jianquan Li, Benyou Wang, Lian Zhang, Ruoyu Sun, Xiang Wan, Haizhou Li, and Jinchao Xu. 2023. Acegpt, localizing large language models in arabic. *ArXiv*, abs/2309.12053.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446.

Zhengbao Jiang, Frank F. Xu, J. Araki, and Graham Neubig. 2019. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

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. *arXiv preprint arXiv:2001.08361*.

Juhao Liang, Zhenyang Cai, Jianqing Zhu, Huang Huang, Kewei Zong, Bang An, Mosen Alharthi, Juncai He, Lian Zhang, Haizhou Li, Benyou Wang, and Jinchao Xu. 2024. Alignment at pre-training! towards native alignment for arabic llms. *ArXiv*, abs/2412.03253.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*.

Haoran Que, Jiaheng Liu, Ge Zhang, Chenchen Zhang, Xingwei Qu, Yi Ma, Feiyu Duan, Zhiqi Bai, Jiakai Wang, Yuanxing Zhang, Xu Tan, Jie Fu, Wenbo Su, Jiamang Wang, Lin Qu, and Bo Zheng. 2024. D-cpt law: Domain-specific continual pre-training scaling law for large language models. *ArXiv*, abs/2406.01375.

Haizhou Shi, Zihao Xu, Hengyi Wang, Weiyi Qin, Wenyuan Wang, Yibin Wang, and Hao Wang. 2024. Continual learning of large language models: A comprehensive survey. *ArXiv*, abs/2404.16789.

Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Xiaodong Song. 2021. Zero-shot information extraction as a unified text-to-triple translation. *ArXiv*, abs/2109.11171.

Chaoyi Wu, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2023. Pmc-llama: Towards building open-source language models for medicine.

Wenzhen Zheng, Wenbo Pan, Xu Xu, Libo Qin, Li Yue, and Ming Zhou. 2024. Breaking language barriers: Cross-lingual continual pre-training at scale. *arXiv preprint arXiv:2407.02118*.

A Details of Data Synthesis

To construct a robust dataset of factual knowledge related to Arabic culture, we first crawled textual content from Wikipedia pages relevant to this domain. We then designed a four-step prompting framework leveraging GPT-4o to extract high-quality triples in the form of (head, relation, tail), as illustrated in Figure 5. Initially, Prompt A was employed to generate a preliminary set of triples, which were often noisy and incomplete. To address this, Prompt B was applied to filter out invalid or nonsensical triples, thereby improving the overall quality of the dataset. Subsequently, Prompt C categorized the relations within the remaining triples, identifying frequently occurring relation types. Given that semantically equivalent relations may exhibit diverse surface forms (e.g., "author" vs. "was written by"), we performed manual refinement to consolidate these variations into a standardized set of relations. Finally, the refined relations were integrated into Prompt D, which was used to re-extract triples from the original web text, resulting in a high-quality set of factual knowledge. This iterative process ensured both precision and consistency in the synthesized data, laying a solid foundation for downstream tasks.

Next, to enhance the diversity and representativeness of the data, we utilized Persona Hub (Ge et al., 2024) along with Prompt E to generate a series of structured templates. For each type of relation, we used the Qwen2.5-72B-Instruct model to generate 1,000 templates, which encompassed a wide range of linguistic expressions. Subsequently, high-quality triples were inserted into these templates to construct a diverse and representative dataset. This dataset not only covers various types of relations but also fully captures complex semantic information, effectively supporting the requirements of downstream tasks.

Furthermore, to meet experimental needs across different scenarios, we adopted a multi-level design strategy during the data synthesis process. Specifically, we generated multiple datasets of varying scales based on different numbers of templates ($N = 10, 50, 100, 200, 500$ and $1,000$) and knowledge point quantities ($K = 500$, 1K, 5K, 50K, and 100K).

B Hyperparameters

The hyperparameters for the experiments are listed in Table 1. For different scales of knowledge point

Prompt C: Triples Classification Prompt

You are a knowledge graph expert. I will provide you with some triples below. These triples involve many categories. Please help me summarize how these triples can be categorized based on their relations. For each category, please output a few of the triples I provided as examples. Place the categories with a higher proportion at the top.

Triples:
{triples}

Figure 6: Triples Classification Prompt: Summary and Examples Based on relation Categories.

Prompt A: Low-Quality Triples Extraction

Extract factual knowledge triples from the text below. Follow these rules:

1. Only include static facts (e.g., dates, authorship, locations).
2. Format each triple as: -[Head Entity | Relationship | Tail Entity], which is equivalent to [Subject | Predicate | Object].
3. Extract at least 20 triples.
4. No explanations needed.

Text:
```
{text}
```

Output format:
```
-[Entity 1 | relationship 1 | Entity 2]
-[Entity 3 | relationship 2 | Entity 4]
```

Figure 7: Prompt for extracting low-quality triples from Wikipedia pages.

quantities ($K = 500$, 1K, 5K, 50K, and 100K), we further optimized the training configuration. When the number of knowledge points was smaller ($K = 0.5$K, 1K, 5K), we adjusted the global batch size from the default 96 to 32, thereby increasing the number of training steps by three times. This adjustment helps the model achieve more sufficient parameter updates and optimization on smaller-scale datasets. For larger-scale knowledge point datasets ($K = 50$K, 100K), we kept the global batch size at 96 to balance training efficiency and model performance.

Hyperparameters	**Value**
Warm-up Steps	0
Gradient Accumulation Steps	2
Max Sequence Length	2048
Learning Rate	7e-6
Min Learning Rate	7e-7
Learning Rate Scheduler	cosine with min lr
Numbers of GPUs	24

Table 1: The list of hyperparameters.

C More Scaling Experiment Results

Prompt B: Triples Filtering Prompt

Analyze whether each extracted triple represents a **unique factual relationship** where the tail entity has no other possible values for the given head entity and relationship. Follow these steps:

1. For each triple, check:

   - If the tail entity **must be unique** (e.g., publication year, locations).
   - Exclude ambiguous relationships (e.g., "crossed by", professions, "travels to", "moved to").
   - Fix incorrect triples by swapping head/tail entities if logically inverted.

2. Examples:

   Invalid

   1. [Wilfred Thesiger | profession | explorer] -> Invalid. "profession" allows multiple values.
   2. [Aziz Nesin | created character | Zübük] -> Invalid. Head/tail inversion because "Zübük" is not the only valid value for the tail entity when head is "Aziz Nesin" and the relation is "created character".
      - Correction: [Zübük | created by | Aziz Nesin], "Aziz Nesin" is the only valid value for the tail entity in this triple.
   3. [The Image Book | Award | Special Palme d'Or] -> Invalid. The Image Book has won more than one award. "Special Palme d'Or" could be replaced by others.
   4. [Brush teeth | timeframe | 8:00 AM] -> Invalid. The entity "Brush teeth" is ambiguous without specifying who performed the brushing.
   5. [J.K. Rowling | wrote | Harry Potter] -> Invalid. Head/tail inversion because "Harry Potter" is not the only valid value for the tail entity when the head is "J.K. Rowling" and the relation is "wrote".
      - Correction: [Harry Potter | written by | J.K. Rowling], "J.K. Rowling" is the only valid value for the tail entity in this triple.
   6. [Mike | travels to | New York] -> Invalid. Mike may travels to other cities, not only "New York" can be the valid value for the tail entity.

   Valid

   1. [Manwakh | located in | Yemen] -> Valid. "located in" is fixed.
   2. [TCP/IP | publication year | 1974] -> Valid. "Publication years" are singular factual events.

3. Analyze each triple below:

Triples to validate:
{triples}

Output format:

Analysis:

1.[Triple 1] → [Valid/Invalid]. *[Brief reason]*.
  - Correction: `[New Head | New Relation | New Tail]` (if applicable)
2.[...]

The Valid/Corrected Triples:
```
-[Head | Relation | Tail]
-[...]
```

Figure 8: Triples Filtering Prompt: Steps and examples for analyzing and verifying unique factual relations. In this process, each triple is examined to determine whether the tail entity is unique for a given head entity and relation, meaning that the tail entity cannot have alternative possible values.

Prompt D: High-Quality Triples Extraction

Please extract triples in the form of (Entity1, Relation, Entity2) from the text provided below. Ensure that each "Relation" is strictly selected from the predefined list of relations provided. If no matching relation can be found in the text based on the predefined list, output `None`.

Predefined Relations:
- **author**: Indicates that Entity2 is the author of Entity1.
  - *Example*: `["Harry Potter", "author", "J.K. Rowling"]` means J.K. Rowling is the author of Harry Potter.
- **director**: Indicates that Entity2 is the director of Entity1.
  - *Example*: `["A", "director", "B"]` means B is the director of A.
- **creater**: Indicates that Entity2 is the creater of Entity1.
- **birth date**: Represents the birth date of Entity1.
  - *Example*: `["Mike", "birth date", "January 1, 1990"]`
- **birth year**: Represents the birth year of Entity1.
  - *Example*: `["Mike", "birth year", "1990"]`
...

Triple Extraction Examples
Text:
```
_**Willow and Wind**_ (Persian: _Beed-o baad_) is a 2000 Iranian drama film directed by Mohammad-Ali Talebi and written by Abbas Kiarostami.

## Cast

* Dariush Afshar as Soraya Esfandiari  *Arman Naderi as Yasmin Khorrami
```

Output:
```json
{
  "triples": [
    ["Willow and Wind", "director", "Mohammad-Ali Talebi"],
    ["Willow and Wind", "author", "Abbas Kiarostami"],
    ["Willow and Wind", "made in", "Iran"],
    ["Willow and Wind", "release year", "2000"],
  ]
}
```

Text for Analysis:
```{content} ```

Please return the results in JSON format as follows:
```json
{
  "triples": [
    [Entity1, relation, Entity2],
    [...]
  ]
}
```

If no triples can be extracted based on the predefined relations, please output:
```json
{
  "triples": null
}
```

Figure 9: High-quality Triples Extraction and Classification: Extracting triples from text based on a predefined list of 26 relation types (partially shown in the figure for brevity). Relations include: B.A. from, Ph.D. from, academic advisor, author, birth city, birth country, birth date, birth year, creator, death date, death year, director, father's name, located in, made in, master's degree from, mother's name, nationality, portrayed by, publish year, publisher, release by, release date, release year, total gross, wife's name. Each extracted triple strictly adheres to this predefined schema.
```

Prompt E: Template Generation Prompt

Assume you are the persona described below, and you are crafting a sentence in the persona's style to describe the relationship between a person and the specific date of their birth.

Requirements:

1.placeholders such as {Head} and {Tail} should be used.
2.The output should be in English.

Persona:
an IT project manager who adopted extreme programming (XP) methodologies on his own team.
Output:
{Head} came into existence on the timeline of life on {Tail}, marking the starting point of their journey.

Persona:
A nature photographer who wants to showcase their stunning photographs with sustainable and unique frames
Output:
{Head} entered the world on the beautiful day of {Tail}, a moment that would inspire a lifetime of capturing nature's splendor.

Persona:
{persona}
Output:

Figure 10: Template Generation Prompt: Generate sentences in the style of a specific person that can be filled with head and tail entities (using the relation between a person and their birth date as an example). To ensure diversity in the generated templates, allow the use of statements similar to the relation in the template for substitution.

Template Examples

Persona: A paralyzed individual who hopes to regain some motor control through brain-computer interface therapy.
Template: {Head} was born on the significant date of {Tail}, initiating a life marked by resilience and the pursuit of groundbreaking advancements in brain-computer interface therapy.

Persona: A blogger who writes in-depth reviews and reflections on each monthly read.
Template: {Head} embarked on their life's narrative on the page of time known as {Tail}, setting the stage for a lifetime of turning the pages of countless stories.

Figure 11: Template Example: Sentences describing entity relations in the style of a specific person (using the relation between a person and their birth date as an example). Fill in the person's name at Head and the birth date at Tail.

Synthesis Data Examples

Data: Saul Bellow was born on the significant date of April 5, 2005, initiating a life marked by resilience and the pursuit of groundbreaking advancements in brain-computer interface therapy.
Head: Saul Bellow
Relation: death date
Tail: April 5, 2005

Data: Bernard Lewis embarked on their life's narrative on the page of time known as May 19 2018, setting the stage for a lifetime of turning the pages of countless stories.
Head: Bernard Lewis
Relation: death date
Tail: May 19, 2018

Figure 12: Synthetic Data Example. (using the relation between a person and their birth date as an example)

Prompt for Evaluation

Task: Predict the tail entity using the given head entity, and relation. Follow the examples below. Output only the tail entity as a noun/noun phrase without articles (e.g., "Paris", not "the Paris"). Use the exact format: "Tail Entity: [answer]".

Example 1:
Head Entity: France
Relation: capital
Tail Entity: Paris

Example 2:
Head Entity: Moby Dick
Relation: author
Tail Entity: Herman Melville

Example 3:
Head Entity: Inception
Relation: director
Tail Entity: Christopher Nolan

Now, predict the tail entity for:
Head Entity: {head}
Relation: {relation}
Tail Entity: {tail}

Figure 13: Prompt for Evaluation.

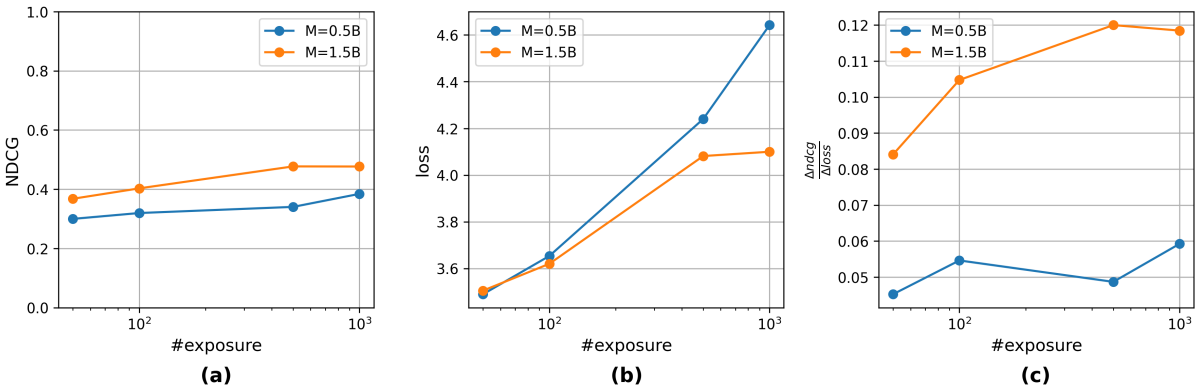

Figure 14: NDCG and loss vs. exposure times with $K = 50,000$

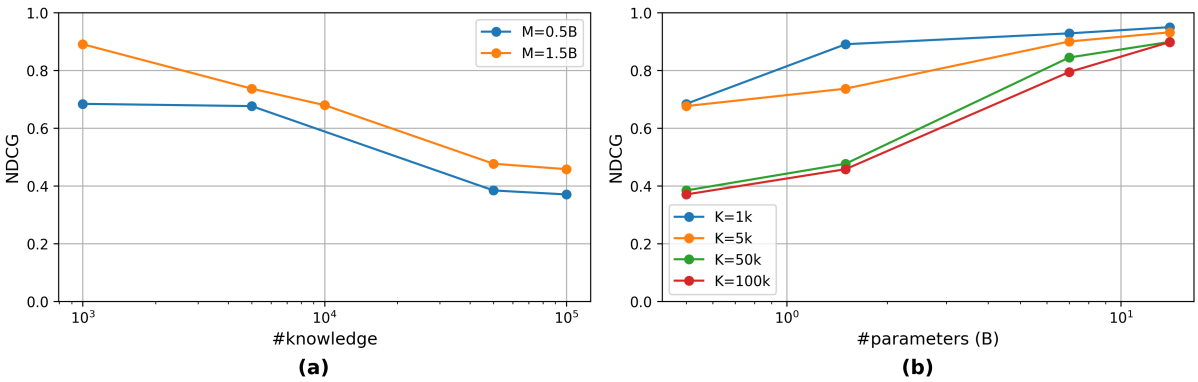

Figure 15: (a): NDCG vs. number of factual knowledge; (b): NDCG vs. number of model parameters. All these curves are obtained with $N = 1,000$.

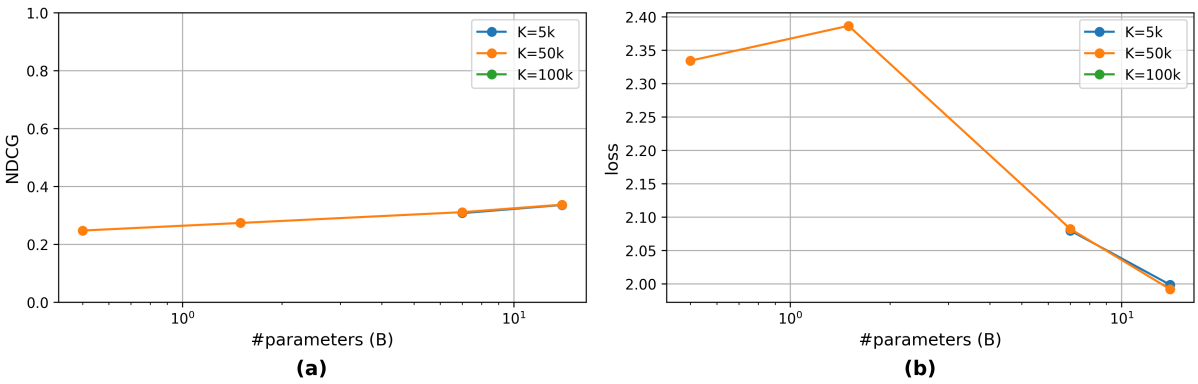

Figure 16: knowledge metrics of pre-trained models; (a): NDCG vs. number model parameters. (b): loss vs. number model parameters with $N = 1,000$.