Enhancing Elusive Clues in Knowledge Learning by Contrasting Attention of Language Models

Jian Gao[2], **Xiao Zhang**[1], **Miao Li**[1,✉], **Ji Wu**[1,3,4]
[1]Department of Electronic Engineering, Tsinghua University
[2]Department of Energy and Power Engineering, Tsinghua University
[3]College of AI, Tsinghua University
[4]Beijing National Research Center for Information Science and Technology
{gaojian21, xzhang19}@mails.tsinghua.edu.cn
{miao-li, wuji_ee}@tsinghua.edu.cn

Abstract

Causal language models acquire vast amount of knowledge from general text corpus during pretraining, but the efficiency of knowledge learning is known to be unsatisfactory, especially when learning from knowledge-dense and small-sized corpora. The deficiency can come from long-distance dependencies which are hard to capture by language models, and overfitting to co-occurrence patterns and distracting clues in the training text. To address these issues, the paper proposes a method to enhance knowledge learning during language model pretraining, by enhancing elusive but important clues in text discovered by the language model themselves. We found that larger language models pay more attention to non-obvious but important clues, which are often overlooked by smaller language models. Therefore, we can identify these clues by contrasting the attention weights of large and small language models. We use the identified clues as a guide to perform token-dropout data augmentation on the training text, and observed a significant boost in both small and large models' performance in fact memorization. This shows that the behavior contrast between more and less-performant language models contains important clues for knowledge learning, and it can be "amplified" for a straight-forward improvement in knowledge learning efficiency.

Code —
https://github.com/tsinghua-msiip/contrasting_attention

Introduction

Pretrained large language models have shown impressive performance on a wide variety of downstream tasks (Ouyang et al. 2022; Chung et al. 2022; Touvron et al. 2023). To achieve good generalization, these models need to be trained on web-scale corpora that are diverse and large enough to capture the complexity of natural language. Unfortunately, it is observed that when training corpora is limited in size or style variation, language models can struggle to generalize the information learned from the corpora (Zhu and Li 2023). This deficiency poses a challenge for injecting knowledge into pretrained language models via continual pretraining (finetuning). In many domains, the available corpora is often limited and knowledge-dense (e.g., in forms of textbooks, manuals, documentations). Such domain text may be difficult to be utilized effectively in finetuning, and the language models may not be able to effectively generalize the domain knowledge to downstream domain tasks.

Not very much is known about the causes of such deficiency in knowledge learning. One likely cause is overfitting to co-occurrence patterns in the limited training text, causing learning of spurious correlations instead of correct factual associations. Another possible reason is the difficulty of capturing long-range dependencies in text, which are crucial for understanding complex relationships. Such deficiency is sometimes a result of intentional design choice in the model architecture, such as the decay of attention weights in the RoPE (Su et al. 2024) positional encodings.

One possible route to understanding this phenomenon is via the attention module in language models. The attention mechanism is a key component that allows the model to focus on different parts of the input when making predictions. The attention weights are shown to be interpretable and explaining the model's behaviors (Clark et al. 2019).

Recently, Yüksekgönül et al. (2023) show that when predicting factual information, models are less likely to attend to the correct clue if the model does not know about the fact. This implies that for new knowledge unknown to the model, the model may not be able to attend to the correct clue at first, leading to difficulty in associating the correct clue (e.g., the head entity) with the prediction target (the tail entity).

To help language models learn, especially smaller models, a common approach is to use knowledge distillation (Hinton, Vinyals, and Dean 2015) (or teacher-student method) to transfer knowledge from a larger model. Given a learning goal, a more performant language model such as GPT-4 (OpenAI 2023) is often used to generate training data for the smaller model (Xu et al. 2024). A main drawback of this approach is that it requires the larger model to be already capable of the task or already have the knowledge. This make it not suitable for learning novel knowledge, such as new facts from an evolving domain. Also, it can only help the smaller model to learn but cannot help the larger model.

In this paper, we propose a simple method to enhance factual knowledge learning in continual pretraining, with the help of a pair of larger and smaller models. Our method is effective in learning novel facts and can boost the performance

of both the larger and smaller models. The main contributions of the paper are as follows:

Attention difference between large and small language models reveals elusive but important clues in text. We show that while large and small language models both show high attention to important and obvious clues in text, large models pay significantly more attention than smaller models to important clues that are less obvious or elusive. Therefore, by contrasting the attention weights of large and small models, we can identify these elusive clues in text that are important for knowledge learning but are often easily overlooked.

Augmenting elusive clues in text boosts knowledge learning in continual pretraining. We show that by using the identified elusive clues as a guide, a token-dropout data augmentation that highlights the elusive clues can significantly boost the model's performance in knowledge learning. We experimented on both synthetic and real-world corpus and show that the proposed method outperforms other forms of data augmentation, and boosting elusive clues universally helps both the large and the small models.

To the best of our knowledge, we are the first to analyze the the attention discrepancies between large and small models and use it for data augmentation. Prior work have distilled attention pattern from large models to small models, but without analyzing what is being distilled. Unlike distillation, our approach also enhances the performance of large models, which is a novel contribution on our part.

We release the code and data used in this paper for reproducibility and further research.

Related Work

Attention as Behavior Explanation

It is observed that attention weights in transformer models provide interpretable clues about the model's behavior. For example, attention heads within multi-head attention can spontaneously differentiate into distinct roles (Clark et al. 2019). Certain heads play a more significant role and affect performance significantly (Voita et al. 2019). More performant models tend to have attention weights that focus more on key information and features, a possible explanation of their superior performance (Yüksekgönül et al. 2023).

Some argue that while attention is somewhat interpretable, its interpretability is not an indicator of model performance (Serrano and Smith 2019). There is divided opinion on the extent to which attention weights reflects true model behavior (Jain and Wallace 2019; Wiegreffe and Pinter 2019). Our study extends these findings by comparing and contrasting attention weights of different models, and show that the difference between attention weights of large and small models can provide important behavioral clues.

Data Augmentation on Text

Data augmentation is a critical technique for enhancing robustness and generalization, especially for limited-size datasets. Various data augmentation methods have been proposed, including random editing of sentences (Wei and Zou 2019) such as insertion, swapping, and deletion. Synonym replacement methods (Mosolova, Fomin, and Bondarenko 2018; Rizos, Hemker, and Schuller 2019) replace words with their synonyms. Contextual augmentation methods (Kobayashi 2018) replace words with other words predicted by a language model for semantic variations. Back-translation (Sennrich, Haddow, and Birch 2016; Edunov et al. 2018) is another commonly used method that generates augmented data by translating to and then back from another language. More sophisticated methods combine multiple augmentations (Xie et al. 2020; Karimi, Rossi, and Prati 2021).

Given that attention provides interpretable clues about the model's behavior, Yu et al. (2022); Hailemariam et al. (2023) uses attention weights to find semantically significant words for replacement augmentation. Lewy and Mandziuk (2023) uses attention weights to find significant input parts for mixup augmentation (Zhang et al. 2018). We go a step further and show that only augmenting the most significant words is insufficient for challenging knowledge learning scenarios, and augmenting hard-to-notice but important parts of the input boosts the model's performance even better than augmenting the significant parts.

Teacher-Student Methods for Language Models

To enhance the performance of smaller models, knowledge distillation methods have been extensively developed to transfer knowledge from larger models to smaller models (Hinton, Vinyals, and Dean 2015; Xu et al. 2024). Large pretrained language models can be used to generate data for finetuning smaller models to transfer its knowledge and skills, for example, instruction following (Wang et al. 2023; Chiang et al. 2023) and reasoning ability (Fu et al. 2023; Ho, Schmid, and Yun 2023). Distillation from large model is also frequently used to build strong domain or task-specific models with a compact size, like for coding (Gunasekar et al. 2023; Rozière et al. 2023) and math (Luo et al. 2023; Yue et al. 2023). Our work explores a different way to utilize large models: we find the behavior difference between large and small models and use it to guide the models towards more difficult part of the text.

Continual Pretraining of Language Models

Continual pretraining takes a language model pretrained on a general corpus and continual the pretraining process with a new corpus, typically domain-specific text, to enhance the model's performance on domain tasks. Model acquires new knowledge and ability via continual pretraining, for example, in coding (Chen et al. 2021), math (Lewkowycz et al. 2022), and medicine (Singhal et al. 2023). We aim at learning new factual knowledge from text via continual pretraining, similar to those in (Jang et al. 2022; Zhu and Li 2023).

Problem Setup: Knowledge Learning Deficiency

Task: Fact Learning in (Continual) Pretraining

Language models can learn factual knowledge from pretraining (or continual pretraining) on text corpora. Zhu and Li

(2023) introduced a synthetic biography dataset for evaluating the efficiency of knowledge learning in language models. The dataset has been utilized by (Khalifa et al. 2024), (Golovneva et al. 2024), and (Saito et al. 2024). It consists of short synthetic biographies of individuals, with a fixed format shown in the following example:

> ***Liam Thompson*** *was born on* ***January 5, 1990****. He spent his early years in* ***Melbourne, Australia****. He received mentorship and guidance from faculty members at* ***Sorbonne University****. He completed his education with a focus on* ***Biomedical Engineering****. He had a professional role at* ***the British Museum****.*

Each biography contains information about an individual's name, birth date, birth city, education, and job status. The task is to finetune (continual pretraining) a language model on the biographies to let it memorize the factual information about the individuals. After training, the model is evaluated on a question-answering task, where we evaluate the model's accuracy in memorizing the underlined part of the biographies.

The questions are formatted like "*When was Liam Thompson born?*". When questions were rephrased using GPT-4, performance generally declined, indicating that the original questioning format yielded the best performance, so that question style has minimal impact on our conclusion.

Deficiency in Knowledge Learning Over Long-Range Dependency

Zhu and Li (2023) have shown that training language models from scratch on the biographies yield poor performance in question answering. We instead perform continual pretraining on pretrained language models up to 70 billion parameters. The language models have undergone extensive pretraining on massive corpora and show strong language capabilities.

We show that even pretrained models with billions of parameters struggle to memorize facts perfectly in continual pretraining. Table 1 show that while Gemma 2 (Team et al. 2024) and LLaMA 3 (Dubey et al. 2024) memorize the first two pieces of information (birth date and birth city) with high accuracy, they struggle to memorize the following three pieces of information (university, major, and company). This rules out the possibility that the performance deficiency is due to limited model size or insufficient pretraining. We also tried swapping the positions of five kinds of information resulted in the same trend: accuracy decreases as distance increases, demonstrating that long-range dependencies, rather than en-tity types, are the primary cause of poor performance.

The performance trend on QA tasks is also plotted in Figure 1. It is clear that as the relationship spans longer distances (i.e., the distance between the tail entity, such as "Company", to the head entity name, the person's name), the model's performance show a decreasing trend. This indicates that the model struggles to capture long-range dependencies in text, which is crucial for learning complex relationships.

One possible reason for the deficiency in learning long-range dependencies is overfitting to a large amount of distracting information between the head and tail entities in a relationship. Overfitting is more likely when relationship only occur in few examples like in the biography dataset. Another possible reason comes from the bias in the model architecture that biases the model's attention towards nearby information. Many popular models, such as LLaMA and Gemma, use the Rotary Position Embedding (RoPE) (Su et al. 2024) as positional encoding in their attention module. RoPE has a long-term decay property, which means that attention weights decay as the relative distance between the key and value token increases. This makes the model focus more on adjacent information but at a cost of important information that are occasionally far-away, hurting the model's performance in learning long-range dependencies.

		Date	City	University	Major	Company
LLaMA 3 8B	EM	0.82	0.91	0.20	0.34	0.09
	F1	0.90	0.93	0.55	0.41	0.11
LLaMA 3 70B	EM	0.98	0.95	0.36	0.73	0.66
	F1	1.00	0.98	0.67	0.77	0.67
Gemma 2 2B	EM	0.98	0.99	0.12	0.54	0.15
	F1	0.98	0.99	0.40	0.57	0.18
Gemma 2 9B	EM	0.99	1.00	0.51	0.89	0.63
	F1	1.00	1.00	0.66	0.90	0.64

Table 1: Performance on the QA task after continual pretraining on the biography corpus.

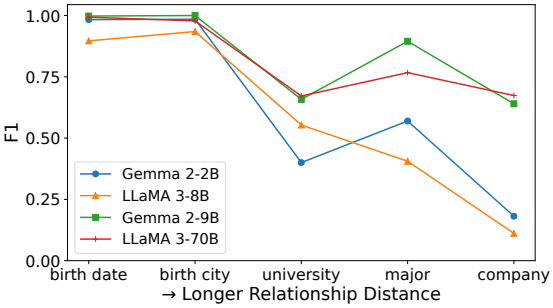


Figure 1: Performance on the QA task show a decreasing trend as the distance between the head and tail entities in the relationship increases in the training text.

Analysis: Contrasting Attention of Language Models

We have shown that language models could achieve near-perfect accuracy in memorizing relationships that span a short distance in text, but struggle when they span a longer distance. In this section, we use attention weights as an interpretability tool to analyze the model's behavior while learning long-range dependencies. We show that LLMs can pay inadequately little attention to key information that is located further away, and more performant larger models can pay more attention to these information than smaller models.

Attention Weight Visualization

We look at model's attention weights to try answering the following question: what information does the model pay attention to when predicting the tail entities in a relationship? The model uses attention weights to retrieve hidden states of context tokens, therefore the weights determines the information flow from the context to the current token in text. Furthermore, if an incorrect head entity is attended to when predicting the tail entity during the forward pass, in backpropagation the model will likely reinforce this incorrect association and cause the model to learn the wrong relationship.

To visualize model's attention weights when predicting the tail entities in a relationship, we extract the attention weights at the *preposition tokens*, i.e., the word immediately preceding the tail entity. For example, in the sentence "He received mentorship and guidance from faculty members *at* Sorbonne University", the attention from the token "*at*" is extracted. Because the model is predicting the tail entity "Sorbonne University" at this position, the attention weights[1] here likely corresponds to the information necessary for predicting it. To ease visualization and for better comparison, instead of directly showing the attention weights, we rank the tokens and visualize the top 10 tokens with the highest attention weights. For each model, we calculate the token attention ranking for 100 biographies[2], and summarize the ranking using a bar plot in Figure 2.

Results show that models assign the most attention to the most important information for predicting the tail entity: the relationship words. The model also pays much attention to the distracting entities in the preceding text. The correct head entity, which is the key information for predicting the tail entity, receives hardly any attention from smaller models and only a small amount of attention from larger models such as Gemma 2 9B, and is almost never ranked in top tokens. This indicates that the model's attention is biased towards short-distance information, which may lead to the model learning the incorrect association and overfitting to such spurious co-occurrences.

Contrasting Attention of Large and Small Language Models

Comparing to smaller models, larger language models tend to have overall better language understanding capabilities, therefore could be more likely to pay attention to the correct clue in the text. For a same family of models, for example, the LLaMA 3 8B and 70B models, the training corpus, model architecture, and training procedure are mostly similar, and they should have relatively similar general behavior pattern besides their capability differences.

Therefore, we can contrast the attention pattern between a large and a small model in the same family to identify the difference in the clue they pay attention to. In Figure 3, we subtract the attention weights of the small model from the large model, and visualize the top 10 tokens with the largest attention differences. The graph shows tokens receiving the most "additional" attention from the large model. It is clear that the correct head entity of the relationship, the "name" tokens (in red color), often receive the most additional attention[3].

Comparing the original model attention in Figure 2 and the attention difference in Figure 3, we can see that while larger models pay more attention to the correct clue in text, the absolute attention weights on the correct clue is still small and biased towards the closer distracting entities. This calls for a method to "amplify" the attention differences so that the model can focus even more on the correct clue in text.

Method: Augmentation From Contrasting Attention

We have shown that important clues that are hard to notice in text can be discovered from the attention difference between large and small models. Next, we propose to utilize and amplify these clues by combining with a simple dropout data augmentation method.

Token-Dropout Data Augmentation

To combat overfitting, token-dropout data augmentation is a simple and effective technique that randomly drops out tokens in a training example (Wei and Zou 2019). Token-dropout introduces noise to the training data and breaks the model's reliance on spurious co-occurrences in the training examples, helping the model achieve better generalization. A naive token-dropout randomly deletes each token independently with a probability α.

Augmentation Guided by Elusive Clues

Although naive token-dropout mitigates overfitting, it does not solve the long-range dependency learning problem. As each token is dropped out independently, the model still suffers from inadequately small attention to non-obvious and distant information. We propose to use the attention difference between large and small models as a guide to dropout tokens in a more selective way. We first use the attention difference to rank the tokens in the training data, and then dropout tokens with a probability that is inversely proportional to their ranking. In this fashion, the model is encouraged to focus more on the tokens containing important but elusive information, as identified by the attention difference.

We use the following function to calculate dropout probability for each token:

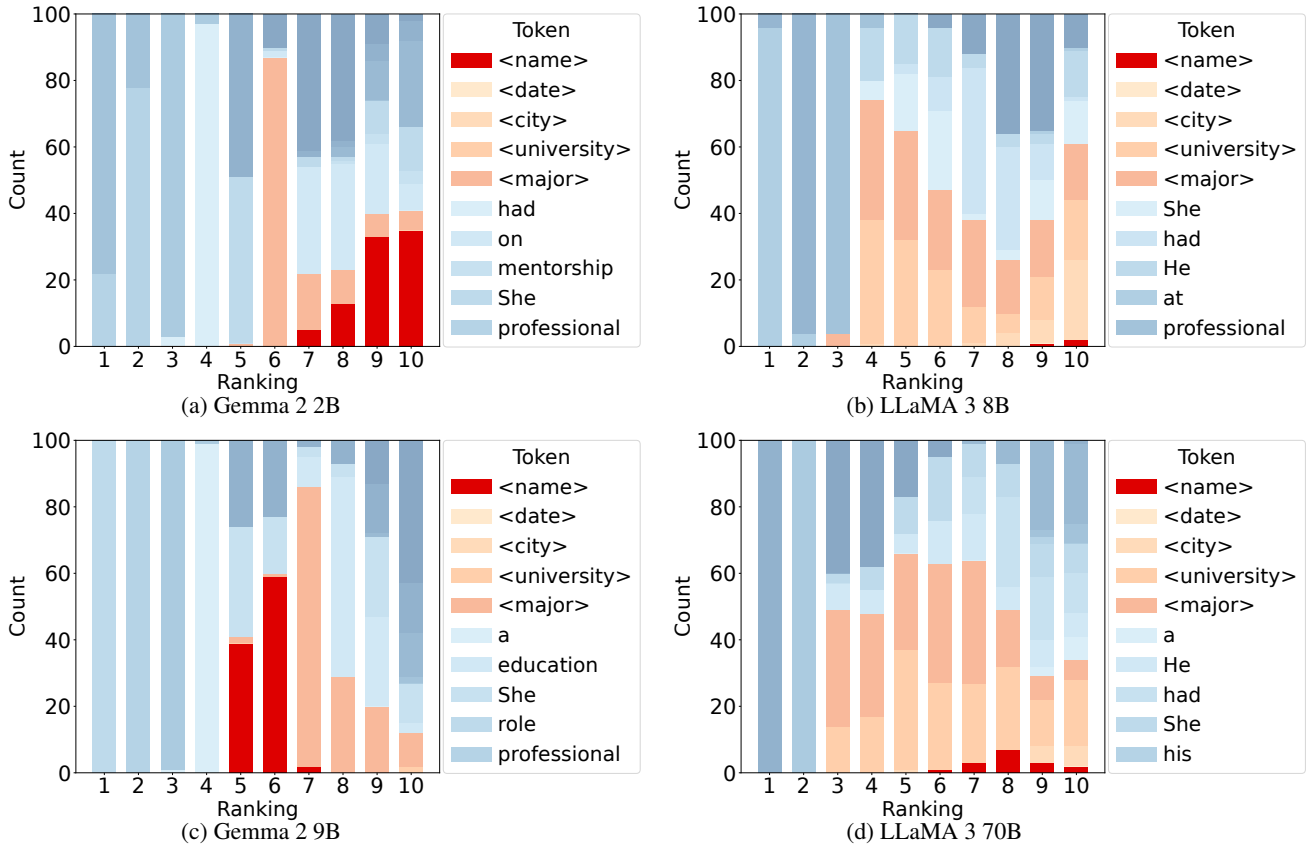
$$p(r) = \alpha(1 - e^{-\beta r}) \tag{1}$$

The token with the r-th rank (having the r-th largest attention difference) will be dropped out with probability $p(r)$. The

[1]To simplify analysis, we took the approach of averaging the attention weights across all layers and attention heads.

[2]Because attention paid on meaningless tokens provides little information, we removed periods, commas, spaces, and placeholders at the beginning of a sentence(for example, <bos>).

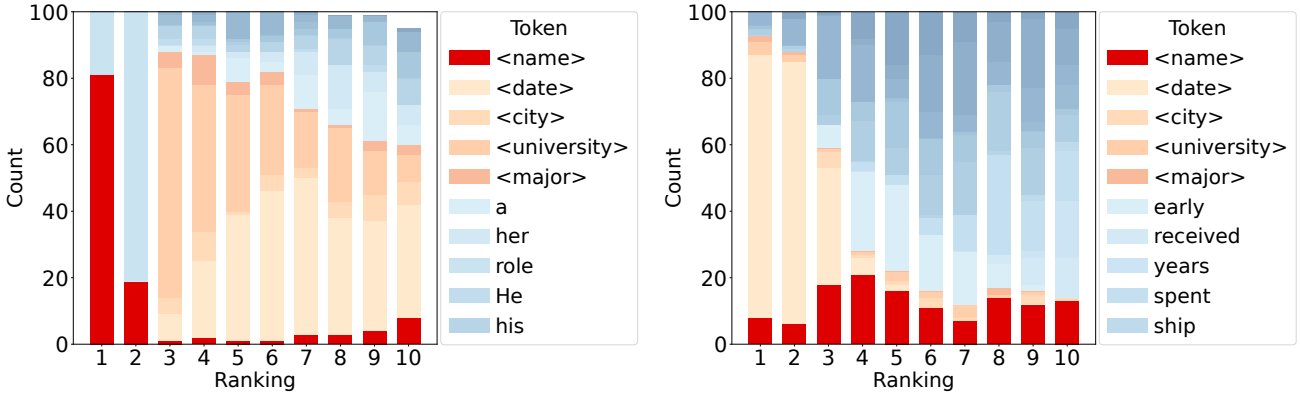
[3]The date tokens also appear to rank high in attention differences, which is likely due to the fact that there are on average more date tokens than name tokens in the text, so they are counted more frequently in the top 10 tokens. For example, under the LLaMA 3 tokenizer, the name is split into an average of 3.56 tokens, while the date is split into around 7 tokens.



(a) Gemma 2 2B (b) LLaMA 3 8B

(c) Gemma 2 9B (d) LLaMA 3 70B

Figure 2: Visualization of tokens receiving the highest attention weights, at the preposition just before the "company" field. Tokens in a sentence are ranked by attention weight, from large to small. Each bar in the graph show the constitution of the i-th ranked token from 100 biographies. "⟨...⟩" denotes tokens belonging to the information fields, and all else are individual tokens. Models generally pay most attention to the relationship words (e.g., "professional", "role", "at"), then to distrating entities in between (e.g., birth date, city, etc.). Because LLaMA 3 models have no special start token at the front of sentences, we add "Text: " at the beginning of sentences to avoid impact of the special position of tokens. All visualization results of LLaMA 3 are done in this way.



(a) Gemma 2 9B/2B (b) LLaMA 3 70B/8B

Figure 3: Visualization of tokens receiving the highest additional attention weights from the large model compared to the small model. For example, the 9B/2B graph visualizes the distribution of the top 10 tokens with the largest *attention_weight(Gemma 2_9B) - attention_weight(Gemma 2_2B)* values. The name tokens (in red), the correct head entity, receive significant additional attention from the larger model.

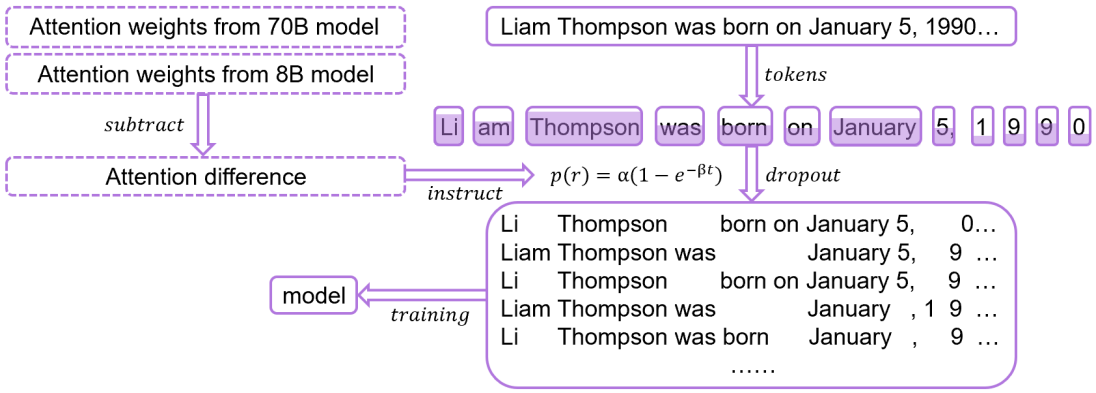


Figure 4: Overview of the proposed data augmentation method based on attention difference between large and small models. Color represents retain probability of each token.

hyperparameter β controls how fast the dropout probability increases with the ranking, and α controls the maximum dropout probability. The tokens with higher attention differences will have lower dropout probabilities, encouraging the model to focus more on these tokens. Figure 4 illustrates the process of the proposed augmentation method.

Results

The Biography Dataset

We use low-rank adaptation (LoRA) (Hu et al. 2022) to facilitate finetuning of models up to 70 billion parameters. As the corpus size is limited, we use a rank of 16 for the LoRA adapters. Adapters are added to all of the model's weights except for the embedding and the output layer. We finetune models with the Huggingface's transformer library (Wolf et al. 2020) on NVIDIA 4090 GPUs. We experiment with LLaMA 3 (Dubey et al. 2024) and Gemma 2 (Team et al. 2024) as two families of language models.

For the baselines, we compare the performance of the models after plain finetuning, random (naive) token-dropout, and token-dropout by attention. In addition to random dropout, dropout by attention uses the original attention weights to guide the dropout probabilities, assuming that the model put more attention on tokens it deemed important. Tokens with lower attention weights are dropped out with higher probabilities to enhance the important information, in a similar vein as in (Yu et al. 2022; Hailemariam et al. 2023). The dropout probabilities are also calculated using Equation 1.

For each experiment, we trained the model from 10 to 30 epochs with learning rates in [5e-5, 1e-3] and selected the model with the best performance. For the augmentation-based methods, we also searched for the best hyperparameters α and β individually for each method. Interestingly, the best hyperparameters for the dropout probabilities happen to be similar for different models and augmentation methods. For each of the augmentation methods, we generate 10 augmented versions of each training example and combine them with the original examples.

Results in Table 2 show that the proposed token-dropout augmentation based on attention difference significantly outperforms other data augmentation methods. We report QA accuracy on the "university" and the "company" fields as the models have poor performance on these fields under plain finetuning (Table 1). We report exact match (EM) accuracy and normalized word-level F1 scores. We can see that while random dropout and dropout by attention improve performance over no data augmentation, our method achieves much more significant improvement. We also collected the results of other information from models trained in our method and accuracy increased across models. This proves that contrasting attention of large and small language models indeed finds important but elusive clues in text effectively, and amplifying these clues in the input has immediate positive effects on the model's memorization efficiency even for the 70B model.

Real-World Dataset

Aside from the biography dataset, we also evaluate the proposed method on Wikipedia text to verify if the method helps knowledge learning on general text. Specifically, we evaluate on the Paragraph-Level Wikipedia Question-Answering dataset (Du and Cardie 2018). We first perform continual pretraining on the Wikipedia text paragraphs (included in the dataset), then evaluate the model's performance on the question-answering data[4]. The questions are specifically designed to incorporate coreference dependencies that span multiple sentences in a paragraph, making it a challenging task that tests the model's ability to learn and memorize complex factual associations.

An example of Wikipedia text from the dataset is:

> *The 2005 edition of the International ISBN Agency's official manual describes how the 13-digit ISBN check digit is calculated. The ISBN-13 check digit, which is the last digit of the ISBN, must range from 0 to 9 and must be such that the sum of all the thirteen digits, each multiplied by its (integer) weight, alternating between 1 and 3, is a multiple of 10.*

[4]This is the "closed-book" setting where the model is not allowed to look at the original Wikipedia passage during question answering. It tests the model's ability to memorize factual knowledge during the continual pretraining phase.

	Hyper-parameters		QA performance University		Company	
	α	β	**EM**	**F1**	**EM**	**F1**
Gemma 2 2B						
Baselines						
Plain finetuning	-	-	0.17	0.48	0.18	0.21
Random token-dropout	0.6	-	0.07	0.38	0.21	0.23
Token-dropout by attention	0.6	0.05	0.19	0.51	0.23	0.29
Ours						
Token-dropout by attention diff	0.6	0.03	**0.25**	**0.56**	**0.32**	**0.36**
Gemma 2 9B						
Baselines						
Plain finetuning	-	-	0.61	0.78	0.63	0.64
Random token-dropout	0.7	-	0.52	0.73	0.51	0.57
Token-dropout by attention	0.6	0.05	0.49	0.62	0.44	0.47
Ours						
Token-dropout by attention diff	0.6	0.03	**0.84**	**0.92**	**0.90**	**0.92**
LLaMA 3 8B						
Baselines						
Plain finetuning	-	-	**0.30**	0.55	0.17	0.21
Random token-dropout	0.6	-	0.11	0.49	0.24	0.29
Token-dropout by attention	0.6	0.05	0.24	0.62	0.21	0.28
Ours						
Token-dropout by attention diff	0.7	0.05	0.29	**0.64**	**0.42**	**0.53**
LLaMA 3 70B						
Baselines						
Plain finetuning	-	-	0.42	0.69	0.66	0.67
Random token-dropout	0.6	-	0.71	0.86	0.71	0.78
Token-dropout by attention	0.7	0.05	0.51	0.75	0.61	0.68
Ours						
Token-dropout by attention diff	0.7	0.01	**0.90**	**0.96**	**0.96**	**0.96**

Table 2: QA performance after continual pretraining on the biography corpus. Data augmentation based on attention difference significantly outperforms other data augmentation methods, for both small and large models.

	Hyper-parameters		QA performance	
	α	β	**EM**	**F1**
Gemma 2 2B				
Baselines				
Plain finetuning	-	-	0.126	0.215
Random token-dropout	0.7	-	0.12	0.223
Token-dropout by attention	0.7	0.005	0.145	0.249
Ours				
Token-dropout by attention diff	0.7	0.005	**0.156**	**0.256**
Gemma 2 9B				
Baselines				
Plain finetuning	-	-	0.186	0.287
Random token-dropout	0.7	-	0.198	0.314
Token-dropout by attention	0.7	0.005	0.205	0.315
Ours				
Token-dropout by attention diff	0.7	0.005	**0.231**	**0.334**
LLaMA 3 8B				
Baselines				
Plain finetuning	-	-	0.146	0.228
Random token-dropout	0.7	-	0.067	0.159
Token-dropout by attention	0.7	0.005	0.134	0.239
Ours				
Token-dropout by attention diff	0.7	0.03	**0.172**	**0.263**
LLaMA 3 70B				
Baselines				
Plain finetuning	-	-	0.179	0.282
Random token-dropout	0.7	-	0.187	0.307
Token-dropout by attention	0.7	0.005	0.190	0.288
Ours				
Token-dropout by attention diff	0.7	0.005	**0.212**	**0.308**

Table 3: QA performance after continual pretraining on the Wikipedia corpus. Data augmentation based on attention difference outperforms other data augmentation methods.

An example of the question from the dataset is as follows:

Question: How many digits does the ISBN have?
Answer: 13

Results in Table 3 show that the proposed method also improves knowledge learning from the Wikipedia text. Unlike naive data augmentation, our method improves the model's memorization efficiency by selectively amplifying difficult and elusive clues. This shows that enhancing the model's focus on important but elusive information is a crucial factor in improving knowledge learning efficiency, and our method is generally applicable to different kinds of text.

Conclusion

Efficiency of learning factual knowledge in not only crucial for pretraining, but also important for effective continual and lifelong learning in language models. Due to the overfitting and long-range dependency problem, even performant language models can struggle to learn and memorize factual knowledge from limited data. In this work, we show that one of the key factors to improving the model's learning, finding the "elusive" but important clues in text, is already embedded in the model's attention weights. However, such clues are hard to discover by the model itself due to the model's bias towards short-range contexts, but clearly manifests themselves when contrasting the attention between a larger and a smaller model. Based on this discovery, we propose a simple yet effective data augmentation method that leverages the attention difference to guide the dropout of tokens in the input. Our method significantly improves the model's performance in memorizing factual knowledge, and is shown to be effective for different corpora and models.

Acknowledgements

This work was supported by the Noncommunicable Chronic Diseases-National Science and Technology Major Project (Grant No. 2023ZD0506501).

References

Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; de Oliveira Pinto, H. P.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; Ray, A.; Puri, R.; Krueger, G.; Petrov, M.; Khlaaf, H.; Sastry, G.; Mishkin, P.; Chan, B.; Gray, S.; Ryder, N.; Pavlov, M.; Power, A.; Kaiser, L.; Bavarian, M.; Winter, C.; Tillet, P.; Such, F. P.; Cummings, D.; Plappert, M.; Chantzis, F.; Barnes, E.; Herbert-Voss, A.; Guss, W. H.; Nichol, A.; Paino, A.; Tezak, N.; Tang, J.; Babuschkin, I.; Balaji, S.; Jain, S.; Saunders, W.; Hesse, C.; Carr, A. N.; Leike, J.; Achiam, J.; Misra, V.; Morikawa, E.; Radford, A.; Knight, M.; Brundage, M.; Murati, M.; Mayer, K.; Welinder, P.; McGrew, B.; Amodei, D.; McCandlish, S.; Sutskever, I.; and Zaremba, W. 2021. Evaluating Large Language Models Trained on Code. *CoRR*, abs/2107.03374.

Chiang, W.-L.; Li, Z.; Lin, Z.; Sheng, Y.; Wu, Z.; Zhang, H.; Zheng, L.; Zhuang, S.; Zhuang, Y.; Gonzalez, J. E.; Stoica, I.; and Xing, E. P. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality.

Chung, H. W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, E.; Wang, X.; Dehghani, M.; Brahma, S.; Webson, A.; Gu, S. S.; Dai, Z.; Suzgun, M.; Chen, X.; Chowdhery, A.; Narang, S.; Mishra, G.; Yu, A.; Zhao, V. Y.; Huang, Y.; Dai, A. M.; Yu, H.; Petrov, S.; Chi, E. H.; Dean, J.; Devlin, J.; Roberts, A.; Zhou, D.; Le, Q. V.; and Wei, J. 2022. Scaling Instruction-Finetuned Language Models. *CoRR*, abs/2210.11416.

Clark, K.; Khandelwal, U.; Levy, O.; and Manning, C. D. 2019. What Does BERT Look At? An Analysis of BERT's Attention. *CoRR*, abs/1906.04341.

Du, X.; and Cardie, C. 2018. Harvesting Paragraph-level Question-Answer Pairs from Wikipedia. In Gurevych, I.; and Miyao, Y., eds., *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1907–1917. Melbourne, Australia: Association for Computational Linguistics.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Edunov, S.; Ott, M.; Auli, M.; and Grangier, D. 2018. Understanding Back-Translation at Scale. In Riloff, E.; Chiang, D.; Hockenmaier, J.; and Tsujii, J., eds., *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, 489–500. Association for Computational Linguistics.

Fu, Y.; Peng, H.; Ou, L.; Sabharwal, A.; and Khot, T. 2023. Specializing Smaller Language Models towards Multi-Step Reasoning. In Krause, A.; Brunskill, E.; Cho, K.; Engelhardt, B.; Sabato, S.; and Scarlett, J., eds., *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, 10421–10430. PMLR.

Golovneva, O.; Allen-Zhu, Z.; Weston, J.; and Sukhbaatar, S. 2024. Reverse Training to Nurse the Reversal Curse. *ArXiv*, abs/2403.13799.

Gunasekar, S.; Zhang, Y.; Aneja, J.; Mendes, C. C. T.; Giorno, A. D.; Gopi, S.; Javaheripi, M.; Kauffmann, P.; de Rosa, G.; Saarikivi, O.; Salim, A.; Shah, S.; Behl, H. S.; Wang, X.; Bubeck, S.; Eldan, R.; Kalai, A. T.; Lee, Y. T.; and Li, Y. 2023. Textbooks Are All You Need. *CoRR*, abs/2306.11644.

Hailemariam, M. Y.; Lynden, S. J.; Matono, A.; and Amagasa, T. 2023. Self-Attention-based Data Augmentation Method for Text Classification. In *Proceedings of the 15th International Conference on Machine Learning and Computing, ICMLC 2023, Zhuhai, China, February 17-20, 2023*, 239–244. ACM.

Hinton, G. E.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. *CoRR*, abs/1503.02531.

Ho, N.; Schmid, L.; and Yun, S. 2023. Large Language Models Are Reasoning Teachers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023*, 14852–14882. Association for Computational Linguistics.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *The Tenth International Conference on Learning Representations, ICLR 2022*. OpenReview.net.

Jain, S.; and Wallace, B. C. 2019. Attention is not Explanation. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, 3543–3556. Association for Computational Linguistics.

Jang, J.; Ye, S.; Yang, S.; Shin, J.; Han, J.; Kim, G.; Choi, S. J.; and Seo, M. 2022. Towards Continual Knowledge Learning of Language Models. In *The Tenth International Conference on Learning Representations, ICLR 2022*. OpenReview.net.

Karimi, A.; Rossi, L.; and Prati, A. 2021. AEDA: An Easier Data Augmentation Technique for Text Classification. In Moens, M.; Huang, X.; Specia, L.; and Yih, S. W., eds., *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, 2748–2754. Association for Computational Linguistics.

Khalifa, M.; Wadden, D.; Strubell, E.; Lee, H.; Wang, L.; Beltagy, I.; and Peng, H. 2024. Source-Aware Training Enables Knowledge Attribution in Language Models. *ArXiv*, abs/2404.01019.

Kobayashi, S. 2018. Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations. In Walker, M. A.; Ji, H.; and Stent, A., eds., *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, 452–457. Association for Computational Linguistics.

Lewkowycz, A.; Andreassen, A.; Dohan, D.; Dyer, E.; Michalewski, H.; Ramasesh, V. V.; Slone, A.; Anil, C.; Schlag, I.; Gutman-Solo, T.; Wu, Y.; Neyshabur, B.; Gur-Ari, G.; and Misra, V. 2022. Solving Quantitative Reasoning Problems with Language Models. In *NeurIPS*.

Lewy, D.; and Mandziuk, J. 2023. AttentionMix: Data augmentation method that relies on BERT attention mechanism. *CoRR*, abs/2309.11104.

Luo, H.; Sun, Q.; Xu, C.; Zhao, P.; Lou, J.; Tao, C.; Geng, X.; Lin, Q.; Chen, S.; and Zhang, D. 2023. WizardMath: Empowering Mathematical Reasoning for Large Language Models via Reinforced Evol-Instruct. *CoRR*, abs/2308.09583.

Mosolova, A.; Fomin, V.; and Bondarenko, I. 2018. Text Augmentation for Neural Networks. In van der Aalst, W. M. P.; Batagelj, V.; Glavas, G.; Ignatov, D. I.; Khachay, M. Y.; Koltsova, O.; Kuznetsov, S. O.; Lomazova, I. A.; Loukachevitch, N. V.; Napoli, A.; Panchenko, A.; Pardalos, P. M.; Pelillo, M.; and Savchenko, A. V., eds., *Supplementary Proceedings of the Seventh International Conference on Analysis of Images, Social Networks and Texts (AIST 2018), Moscow, Russia, July 5 - 7, 2018*, volume 2268 of *CEUR Workshop Proceedings*, 104–109. CEUR-WS.org.

OpenAI. 2023. GPT-4 Technical Report. Technical report.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C. L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; Schulman,

J.; Hilton, J.; Kelton, F.; Miller, L.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P. F.; Leike, J.; and Lowe, R. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

Rizos, G.; Hemker, K.; and Schuller, B. W. 2019. Augment to Prevent: Short-Text Data Augmentation in Deep Learning for Hate-Speech Classification. In Zhu, W.; Tao, D.; Cheng, X.; Cui, P.; Rundensteiner, E. A.; Carmel, D.; He, Q.; and Yu, J. X., eds., *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, 991–1000. ACM.

Rozière, B.; Gehring, J.; Gloeckle, F.; Sootla, S.; Gat, I.; Tan, X. E.; Adi, Y.; Liu, J.; Remez, T.; Rapin, J.; Kozhevnikov, A.; Evtimov, I.; Bitton, J.; Bhatt, M.; Canton-Ferrer, C.; Grattafiori, A.; Xiong, W.; Défossez, A.; Copet, J.; Azhar, F.; Touvron, H.; Martin, L.; Usunier, N.; Scialom, T.; and Synnaeve, G. 2023. Code Llama: Open Foundation Models for Code. *CoRR*, abs/2308.12950.

Saito, K.; Sohn, K.; Lee, C.-Y.; and Ushiku, Y. 2024. Where is the answer? Investigating Positional Bias in Language Model Knowledge Extraction.

Sennrich, R.; Haddow, B.; and Birch, A. 2016. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Serrano, S.; and Smith, N. A. 2019. Is Attention Interpretable? In Korhonen, A.; Traum, D. R.; and Màrquez, L., eds., *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, 2931–2951. Association for Computational Linguistics.

Singhal, K.; Tu, T.; Gottweis, J.; Sayres, R.; Wulczyn, E.; Hou, L.; Clark, K.; Pfohl, S.; Cole-Lewis, H.; Neal, D.; Schaekermann, M.; Wang, A.; Amin, M.; Lachgar, S.; Mansfield, P. A.; Prakash, S.; Green, B.; Dominowska, E.; y Arcas, B. A.; Tomasev, N.; Liu, Y.; Wong, R.; Semturs, C.; Mahdavi, S. S.; Barral, J. K.; Webster, D. R.; Corrado, G. S.; Matias, Y.; Azizi, S.; Karthikesalingam, A.; and Natarajan, V. 2023. Towards Expert-Level Medical Question Answering with Large Language Models. *CoRR*, abs/2305.09617.

Su, J.; Ahmed, M. H. M.; Lu, Y.; Pan, S.; Bo, W.; and Liu, Y. 2024. RoFormer: Enhanced transformer with Rotary Position Embedding. *Neurocomputing*, 568: 127063.

Team, G.; Riviere, M.; Pathak, S.; Sessa, P. G.; Hardin, C.; Bhupatiraju, S.; Hussenot, L.; Mesnard, T.; Shahriari, B.; Ramé, A.; et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; Rodriguez, A.; Joulin, A.; Grave, E.; and Lample, G. 2023. LLaMA: Open and Efficient Foundation Language Models. *CoRR*, abs/2302.13971.

Voita, E.; Talbot, D.; Moiseev, F.; Sennrich, R.; and Titov, I. 2019. Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned. *CoRR*, abs/1905.09418.

Wang, Y.; Kordi, Y.; Mishra, S.; Liu, A.; Smith, N. A.; Khashabi, D.; and Hajishirzi, H. 2023. Self-Instruct: Aligning Language Models with Self-Generated Instructions. In Rogers, A.; Boyd-Graber, J. L.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, 13484–13508. Association for Computational Linguistics.

Wei, J. W.; and Zou, K. 2019. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, 6381–6387. Association for Computational Linguistics.

Wiegreffe, S.; and Pinter, Y. 2019. Attention is not not Explanation. In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, 11–20. Association for Computational Linguistics.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; Davison, J.; Shleifer, S.; von Platen, P.; Ma, C.; Jernite, Y.; Plu, J.; Xu, C.; Scao, T. L.; Gugger, S.; Drame, M.; Lhoest, Q.; and Rush, A. M. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos*, 38–45. Association for Computational Linguistics.

Xie, Q.; Dai, Z.; Hovy, E. H.; Luong, T.; and Le, Q. 2020. Unsupervised Data Augmentation for Consistency Training. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Xu, X.; Li, M.; Tao, C.; Shen, T.; Cheng, R.; Li, J.; Xu, C.; Tao, D.; and Zhou, T. 2024. A Survey on Knowledge Distillation of Large Language Models. *CoRR*, abs/2402.13116.

Yu, Y. J.; Yoon, S. J.; Jun, S. Y.; and Kim, J. W. 2022. TABAS: Text augmentation based on attention score for text classification model. *ICT Express*, 8(4): 549–554.

Yue, X.; Qu, X.; Zhang, G.; Fu, Y.; Huang, W.; Sun, H.; Su, Y.; and Chen, W. 2023. MAmmoTH: Building Math Generalist Models through Hybrid Instruction Tuning. *CoRR*, abs/2309.05653.

Yüksekgönül, M.; Chandrasekaran, V.; Jones, E.; Gunasekar, S.; Naik, R.; Palangi, H.; Kamar, E.; and Nushi, B. 2023. Attention Satisfies: A Constraint-Satisfaction Lens on Factual Errors of Language Models. *CoRR*, abs/2309.15098.

Zhang, H.; Cissé, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2018. mixup: Beyond Empirical Risk Minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Zhu, Z. A.; and Li, Y. 2023. Physics of Language Models: Part 3.1, Knowledge Storage and Extraction. *CoRR*, abs/2309.14316.