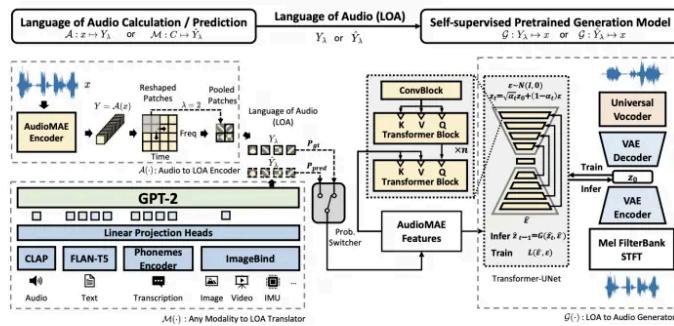


# Review 121, Short: AUDIOLDM 2: LEARNING HOLISTIC AUDIO GENERATION WITH SELF-SUPERVISED PRETRAINING, 11.08.23

<https://huggingface.co/papers/2308.05734>



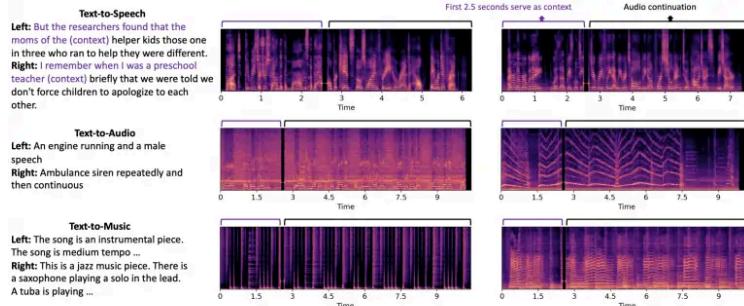
**Figure 1:** The overview of the *AudioLDM 2* architecture. The AudioMAE feature is a proxy that bridges the audio semantic language model stage (GPT-2) and the semantic reconstruction stage (latent diffusion model). The probabilistic switcher controls the probability of the latent diffusion model using the ground truth AudioMAE ( $P_{gt}$ ) and the GPT-2 generated AudioMAE feature ( $P_{pred}$ ) as the condition. Both the AudioMAE and latent diffusion models are self-supervised pre-trained with audio data.

רוצים לגנרט/לעורך את אודיו מגוון סוגים של דатаה? מתרבר שnitan ליצור את אודיו מתמונה, וידאו, טקסט ואפילו את IMU (המתאר אופייני תנוצה של אובייקט כמו מהירות כיוונית ותאוצה).

היום ב-shorthbrewpaperreviews# סוקרים מאמר שמציע מודל המסוגל לעשות את זה. בגדול במשימות מולטימדליות עיקרי הבעיה היא למפות את כל סוג הדadataה למרחב השיכון(embedding) באופן הגיוני. לעומת השיכון של אודיו של ציוץ ציפורים צריך להיות קרוב לשיכון של טקסט "ציוץ של ציפורים". המחברים משתמשים במודלים מאומנים עבור כל סוג הדadataה שהם יוצרים אודיו כדי למפות את הדadataה למרחב השיכון (למשל לVIDAO ו-*ImeBind* המשמשים ב-*T5*, *Flan-T5*, *ImageBind*, ו-*CLAP*).

לאחר מכון הם מכילים 2-*gpt* שמטרתו להעביר את כל הייצוגים לאוטו מרחב השיכון. לאחר מכן לוקחים את הפלט של 2-*gpt* ומעבירים אותו למודל דיפוזיה לטוני המאומן לגנרט אודיו בהתאם לתנאי (המקודד באמצעות וקטור השיכון). את מודל הדיפוזיה הלטוני מאומנים על דטה אודיו לא מותג כאשר לפני כניסה לאנקודר מעבירים אותו לتردد מל (mel frequencies).

לאחר מכון מכילים את המודל עם הדטה המותג (אודיו והתיאור שלו). פרט נוסף מעניין לגבי המאמר הוא שבמהלך האימון ב-75% מהמרקם משתמשים בשיכון של התיאור כדי לגנרט אודיו וב-25% הנורטירם מכנים במוקומו את הייצוג של האודיו העצמו המופק באמצעות מודל AudioMAE (אנקודר של אוטו אודיו).



**Figure 3:** In-context learning ability of *AudioLDM 2*. The left column shows the ground truth audio, where the leading 2.5 seconds are used as context for audio generation. The continuation of the audio context is shown in the right column. We manually insert a 0.15 seconds beep sound before the continuation for better demonstration. More demo is available in Figure 4, Figure 5, and Figure 9.

## Review 122, Short: DIVIDE & BIND YOUR ATTENTION FOR IMPROVED GENERATIVE SEMANTIC NURSING, 12.08.23

<https://huggingface.co/papers/2307.10864>

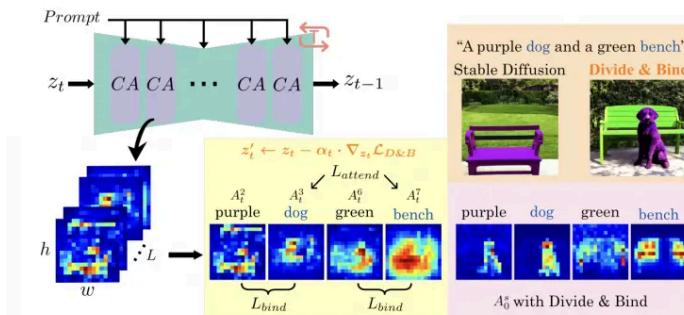


Figure 2: Method overview. We perform latent optimization on-the-fly based on the attention maps of the object tokens with our TV-based  $L_{attend}$  and JSD-based  $L_{bind}$ .

בוח שמתם לב שלפעמים שאתם מבקשים ממודל גנרטיבי (ג'יד MidJourney) ליצור תמונה עם כמה אובייקטים (ג'יד חתול, כלב, שולחן ומקרר) התמונה לא תמיד מכילה את כל האובייקטים במיוחד כאשר התיאור הוא די ארוך ומכל מספר רב של אובייקטים.

היום ב-#shortherebrewpaperreviews סוקרים מאמר המציג מענה לשוגיה זו. קודם כל נבין למה לא תמיד אנו מצליחים להעיר למודל גנרטיבי (מודל דיפוזיה) את כל המידע. כדי להבין זאת נזכיר שהמודע (יצוג) של הטוקנים מזון למנגנון cross-attention לתוכן השכבות הפנימיות של רשת UNet שהיא הלב של מודל הדיפוזיה (מערכת את הרוש בערך איטרציה).

از לפעמים טוין טקסטואלי אחד "גונב" את כל "תשומת הלב" ואז הטוקנים האחרים פשוט לא באים לידי ביטוי ונעלמים מההתמונה. שוגיה נוספת שעלולה לקרות כאן היא attribute binding שבו פיצ'רים של אובייקטים מסוימים (כגון צבע או טקסטורה) משוכרים לאובייקטים אחרים. המחברים מציעים להתמודד עם סוגיות אלו עם שתי גישות חדשות ל"הזהה" של ייצוג הדאטה בכל איטרציה (semantic guiding) לכיוון של גרדיאנט הפונקציה המנסה לאכוף תכונות רצויות של מופות ה-*attention*. קודם כל הוא מנסה לכפות שונות מקסימלית בו מופות

ה-attention של כל אובייקט בין טוקנים ויזואליים סמוכים (פאיים של תמונה). לטענת המחברים ככה מונעים מאובייקט אחד להשתלט לנו על כל הטוקנים הויזואליים.

התוכונה השנייה שמנסם לאכוף היא שונה בין מיפות attention של אובייקטים שונים. את זה הם משיגים עם מקסום של מרחק (JSD) Jensen-Shannon בין מיפות ה-attention המנורמלות בין כל שני האובייקטים. כאמור מציגים את שערוך הייצוג של כל איטרציה של מודל דיפוזיה בכיוון שסוכם מנורמל את שני ה"יעדים" האלה.



## Review 123, Short: TextDiffuser: Diffusion Models as Text Painters, 13.08.23

<https://huggingface.co/papers/2305.10855>

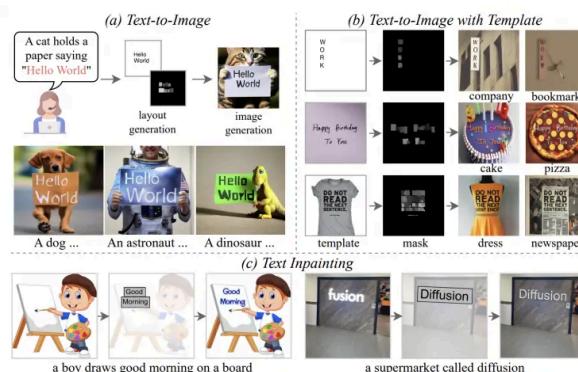


Figure 1: TextDiffuser generates accurate and coherent text images from text prompts or together with template images, as well as conducting text inpainting to reconstruct incomplete images.

מודלי דיפוזיה מצטיינים ביצירת תמונות מרישיות מתיאור טקסטואלי אך עדין מתקשות ביצירה תמונות המכילות טקסט כחלק מהתמונה. למשל יצירת תמונה של כלב המחזיק שלט שכותב עליו "ברוך הבא הביתה" עלולה ליצור תמונה עם טקסט שונה ולא ברור על השלט.

היום ב [shorthebrewpaperreviews](#) סקררים מאמר שמנסה לתת מענה לסוגיה זו. המאמר מציע גישה דו-שלבית שבשלב הראשון נוצרת תמונה שבה נוצרת את החלק בתמונה המכיל טקסט ובשלב השני מלבשים על

התמונה זו את האובייקטים שיש בתמונה ושאר הפרטים (כגון טקסטורה ורקע). בשלב הראשון קודם כל בונים את שיכון (embedding) של הטקסט עם CLIP מאומן.

אך להבדיל ממודלי דיפוזיה גנרטיביים אחרים מוסיפים לוקטור השיכון מושגים לכך שהיכון נלמד של מילוט המפתח (מחלקים את התיאור למילים שצרכות להופיע בתמונה ואלה שלא ובונם וקטורי שיכון שלהם). בנוסף מוסיפים לוקטור השיכון קידוד נלמד של רוחב של כל אות בתמונה ובנוסף מוסיפים לכך קידוד תלויות מיקום (positional encoding) נלמד. כל השיכונים הללו מחושבים באמצעות שני טרנספורמרים: אנקודר ודקדור.

הראשון מאומן לקודד את הדאטה, השני מחשב (BB) Bounding Boxes עבור האותיות בתמונה ובשלב האחרון מрендרים את התמונה לפי ה-BB שהি�ישנו והאותיות (המקודדות) ומגנרטים מסכותאותיות. בשלב השני קודם כל מגנרטים תמונה כאשר הקטל הוא מסכות הנוצרות בשלב הראשון (בכמה צורות), השיכון של הטקסט והתמונה המורעשת (הרי זה מודל דיפוזיה). בנוסף ללוס הריגל של מודל הדיפוזיה המודל נכנס על אי התאמת של מיקום האותיות בתמונה (הם מאמנים רשות לזיהוי מיקומים אלה).

## Review 124, Short: Self-Alignment with Instruction Backtranslation, 14.08.23

<https://huggingface.co/papers/2308.06259>

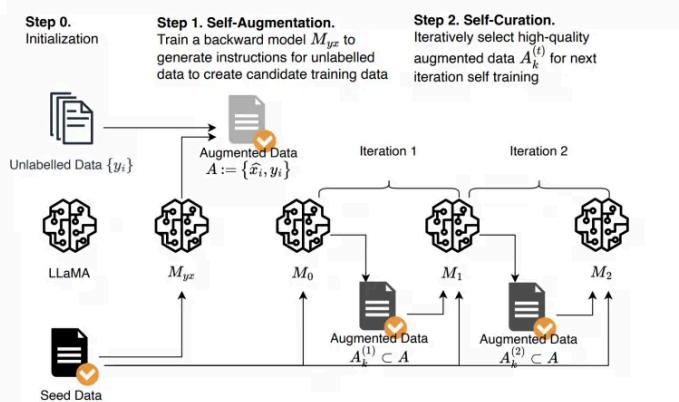


Figure 1: An overview of our **instruction backtranslation** method. We start from a base language model, e.g. LLaMa, a small amount of seed examples of (instruction, output) pairs, and a collection of unlabelled documents which are considered candidate outputs for unknown instructions. **Self-augmentation:** the base model is finetuned with (output, instruction) pairs from the seed examples as an instruction prediction model  $M_{yx}$ , which is used to generate candidate instructions for outputs from the unlabelled data. **Self-curation:** starting from an intermediate instruction-following model

נניח שאתם רוצים לבצע יישור (alignment) עם DATA מתויג (instruction tuning) של מודל השפה שלכם שאימנתם קודם על DATA גדול ולא מתויג. נניח שיש בידיכם DATA מתויג איזוטי לא גדול במיוחד וDATAstein מאד גדול ומוגן אך לא מתויג.

המאמר שנסקו היום ב-wsorshethebrewpaperreviews מציע שיטה אינטואיטיבית ואלגנטית להפקה של DATAstein איזוטי מתויג בגודל ממשועורי מהDATAstein הלא מתויג שיש ברשותנו. תהליך האימון מורכב משני שלבים עיקריים: קודם כל מכילים מודל מאומן ליצור הוראה (instruction) מההשובה עם DATAstein האיזוטי המתויג שיש לנו.

לאחר מכן מזינים למודל את ה"תשובות" מהדאטסט הלא מתואג כדי ליצור הוראה לכל לכל תשובה. שלב זה נקרא self-augmentation. כמובן שלא כל הזוגות שיצרנו הם באיכות גבוהה וכן מפלטים אותם בשלב השני הנקרא self-curation. לוקחים מודל שמאכיל רק עם הדוגמאות מהדאטסט המתואג איכותית (הקטן). מבקשים את המודל (עם פרומפט ספציפי) לדרג מ-1 עד 5 את התאמת התשובה להוראה.

לאחר מכן מפלטים את הזוגות בעלי ציונים הנמוכים. המחברים גם הציעו מה שנקרא iterative self-curation שבמהלכו לוקחים את הזוגות (הוראה, תשובה) בעלי ציונים גבוהים, מכילים את המודל עם זה. לאחר מכן ניתן למנף את המודל-self-augmentation (השלב הראשון) כדי ליצור דאטסט מתואג איכות יותר. ניתן לחזור על התהילה כמה פעמים בתקווה לקבל דאטסט מתואג גדול ו איכותי. מאמר אלגנטי ונחמד ...

Below is an instruction from an user and a candidate answer. Evaluate whether or not the answer is a good example of how AI Assistant should respond to the user's instruction. Please assign a score using the following 5-point scale:  
1: It means the answer is incomplete, vague, off-topic, controversial, or not exactly what the user asked for. For example, some content seems missing, numbered list does not start from the beginning, the opening sentence repeats user's question. Or the response is from another person's perspective with their personal experience (e.g. taken from blog posts), or looks like an answer from a forum. Or it contains promotional text, navigation text, or other irrelevant information.  
2: It means the answer addresses most of the asks from the user. It does not directly address the user's question. For example, it only provides a high-level methodology instead of the exact solution to user's question.  
3: It means the answer is helpful but not written by an AI Assistant. It addresses all the basic asks from the user. It is complete and self-contained with the drawback that the response is not written from an AI assistant's perspective, but from other people's perspective. The content looks like an excerpt from a blog post, web page, or web search results. For example, it contains personal experience or opinion, mentions comments section, or share on social media, etc.  
4: It means the answer is written from an AI assistant's perspective with a clear focus of addressing the instruction. It provides a complete, clear, and comprehensive response to user's question or instruction without missing or irrelevant information. It is well organized, self-contained, and written in a helpful tone. It has minor room for improvement, e.g. more concise and focused.  
5: It means it is a perfect answer from an AI Assistant. It has a clear focus on being a helpful AI Assistant, where the response looks like intentionally written to address the user's question or instruction without any irrelevant sentences. The answer provides high quality content, demonstrating expert knowledge in the area, is very well written, logical, easy-to-follow, engaging and insightful.  
Please first provide a brief reasoning you used to derive the rating score, and then write "Score: <rating>" in the last line.  
<generated instruction>  
<output>

## Review 125 : Convolutions Die Hard: Open-Vocabulary Segmentation with Single Frozen Convolutional CLIP, 15.08.23

<https://huggingface.co/papers/2308.02487>



Figure 1: **k-means visualization on top of frozen CLIP backbone features w.r.t. different input resolutions.** Both ViT-based and CNN-based CLIP produces semantic-meaningful features. However, when scaling up the input resolutions, we note that ViT-based CLIP features turn noisier, while CNN-based ones are smoother and generalize better. The smoother feature map is preferable for mask-pooling modules in our design.

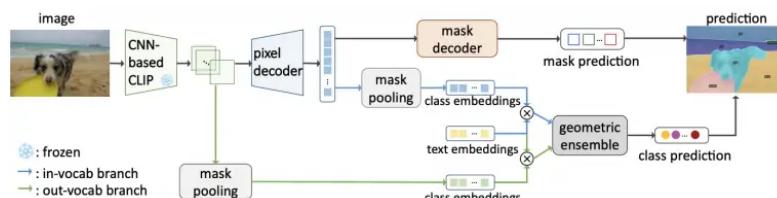
בד"כ סגמנטציה בתמונות מותבצעת ב-2 שלבים. קודם מחשבים מסכות עבור כל האובייקטים בתמונה ו בשלב השני מזהים סוג האובייקטים. בזמן האחרון יש שימוש רב במודלים מאומנים (כמו CLIP) להפקה של ייצוג התמונה; בשלב 1 מזינים את התמונה למודל המאומן ובשלב 2 מזינים אותה יחד עם המסכות.

היום ב#shorthebrewpapereviews סוקרים מאמר המבצע זאת בשלב אחד. למה זה טוב בעצם? כי במקרה זה צריך להזין את התמונה ל-CLIP רק פעם אחת שזה מזכיר משמעותית את זמן ההසקה והאימון כי CLIP זה מודל גדול ונכבד. איך הם עושים זאת?

קדם כל מעבירים את התמונה דרך CLIP מוקפא ואז מזינים את הייצוג המופק אליו ל-Pixel Decoder יחד עם "שאלות האובייקטים" (סוג של פרומפט לחיפוש האובייקט) לחיזוי המסכות. במהלך האימון מבצעים התאמת בין המסכות ground-truth לבין המסכות שהוציאו באמצעות אלגוריתם התאמת הונגרי (mezogim מסכות הדומות ביותר). לאחר שבנוינו את המסכות אנו צריכים לזרזות את התוכן בתוך המסכות שמצאנו.

עבור סגמנטציה עם מיליון סגור (הקטגוריות ידועות) מצליבים את הייצוג (שיכון) של הקטgorיה במסכה (המופק באמצעות הפעלת רשת pooling שלpixel decoder) על הפלט של CLIP (עם ייצוג הטקסט) של כל קטgorיה במטרה למצאו קטgorיה הטובה ביותר לכל מסכה. זה נעשה באמצעות חישוב דמיון cosine (עם טמפרטורה נלמדת) בין ייצוגים אלו כאשר קטgorיה עם דמיון מקסימלי עם ייצוג נבחרת קטgorיה של המסכה.

כדי לאפשר אוגמנטציה עם מיליון פתוח (עבור קטgorיות שלא אומנו במהלך האימון) המחברים יוצרים ייצוג המסכה (מוסיפים "נתיב" למודל המקורי לנוטיב המילון הסגור) רק מהשיכון המופק מה贊ת התמונה ל-CLIP (דרך mask pooling). ואז ב-*inference* משתמשים באותה שיטה שהסבירנו עבור המילון הסגור. בשלב משלבים את החיזוי עבור המילון הסגור והפתחו דרך *geometric ensemble* (סוג של מיזוג).



## Review 126, Short : SOLVING CHALLENGING MATH WORD PROBLEMS USING GPT-4 CODE INTERPRETER WITH CODE-BASED SELF-VERIFICATION, 16.08.23

<https://huggingface.co/papers/2308.07921>

**Question**

**Question:**  
The greatest common divisor of positive integers  $m$  and  $n$  is 6. The least common multiple of  $m$  and  $n$  is 126. What is the least possible value of  $m + n$ ? **Answer:** 60

**(c) Model output**

**BasicPrompt**  
Solve the problem and put your answer in \boxed{ }. The problem is: {Question}.

One important property relating greatest common divisor (GCD) and least common multiple (LCM) is that  $\text{GCD}(m, n) \times \text{LCM}(m, n) = m \times n$

Given in the problem, we can get  $6 \times 126 = m \times n$

Let's compute the right hand side of the equation first.

```
code
gcd,lcm = 6, 126
gcd * lcm
>>> 756
```

The product of  $m$  and  $n$  is 756. Since the GCD of  $m$  and  $n$  is 6, both  $m$  and  $n$  must be multiples of 6. Let's find all the pairs of multiples of 6 that multiply to 756. We'll do this by finding all the factors of 756 that are multiples of 6.

```
code
[(), m.times.n // i for i in range(6, int(m.times.n**0.5) + 1, 6) if m.times.n % i == 0]
>>> [(6, 126), (12, 63), (18, 42)]
```

The pairs of factors of 756 that are multiples of 6 are: 1. (6, 126) 2. (12, 63) 3. (18, 42)  
We want to find the pair with the least sum. Let's compute the sums for each pair, identify the smallest one.

```
code
sums = [(pair, sum(pair)) for pair in factor.pairs]
min(sums, key=lambda x: x[1])
>>> ((18, 42), 60)
```

The pair with the smallest sum is (18, 42), and the sum is 60. So, the least possible value is 60 ✓

**(a) Model output**

**Prompt 1**  
You are not allowed to use any code. The problem is: {Question}.

We know that for any two positive integers  $m$  and  $n$ , the product of their greatest common divisor (GCD) and least common multiple (LCM) is equal to the product of the two numbers. This can be written as:

$$\text{GCD}(m, n) \cdot \text{LCM}(m, n) = m \cdot n$$

Given that  $\text{GCD}(m, n) = 6$  and  $\text{LCM}(m, n) = 126$ , we can substitute these values into the equation:

$$6 \cdot 126 = m \cdot n$$

Solving this, gives us  $m \cdot n = 756$ . Now, we need to find the pair of factors of 756 that add up to the smallest sum, as we want to find the least possible value of  $m + n$ .

The pairs of factors of 756 are (1,756), (2,378), (3,252), (4,189), (6,126), (7,108), (9,84), (12,63), (14,54), (18,42), (21,36), (27,28). We can see that the pair (21,36) gives the smallest sum, which is  $21 + 36 = 57$ . So, the least possible value of  $m + n$  is 57.

Therefore, the solution to the problem is 57 ✗

**(b) Model output**

**Prompt 2**  
You can use code only 1 time. The problem is: {Question}.

First, we should recall that for any two positive integers  $m$  and  $n$ , their greatest common divisor (gcd) and least common multiple (lcm) satisfy the relationship:

$$m \cdot n = \text{gcd}(m, n) \cdot \text{lcm}(m, n)$$

In this problem, we're given that  $\text{gcd}(m, n) = 6$  and  $\text{lcm}(m, n) = 126$ . We can substitute these values into the formula to get:

$$m \cdot n = 6 \cdot 126$$

We can solve this equation for one variable and then use the fact that  $m$  and  $n$  are integers to generate possible pairs  $(m, n)$ . After we have the pairs  $(m, n)$ , we can find the pair that minimizes  $m + n$ .

```
code
import sympy as sp
gcd,lcm = 6,126
product = gcd * lcm
pairs = [(d, product // d) for d in sp.divisors(product) if d <= product // d]
min_pair = min(pairs, key=lambda x: x[0] + x[1])
min_pair[0], min_pair[1]
```

The pair  $(m, n)$  that minimizes  $m + n$  is (27, 28), and the minimum possible value of  $m + n$  is 55 ✗

אתם בטוח שמתם לב למודל שפה מסוגלים לתקן את ה"שגיאות" שהם עושים אם מוצבאים להם עלייהם. כמובן אם מודל שפה ענה לא נכון לפעמים מתקן את עצמו אם מבקשים ממנו לבדוק את תשובתו (לפעמים הוא "מתקן" גם תשובות נכונות אם מטילים בהם ספק). אבל מה אם נבקש ממנו לבדוק את תשובות דרך מימוש בקוד.

היום ב-#shorthebrewpapereviews סוקרים מאמר שמציע שיטה לפתור בעיות מתמטיות על ידי מודל שפה באמצעות מימוש בקוד. הרעיון של המאמר הינו מאוד אינטואיטיבי ואלגנטiy. מבקשים מודל שפה לפתור בעיה מתמטית שלב אחריו שלב תוך כדי שימוש ב-code interpreter ובנוסף מבקשים לוודא את התשובה עם אותו code interpreter. כמובן המודל כותב קוד הנחוץ לפתור בעיה, מודיא שהפתרון נכון בעזרת הקוד ואם זה לא, הוא שוב כותב קוד המתקן את הטעות ושוב בודק אותה (לא הצלחתי להבין האם פעולות אלו חוזרות על עצמן יותר מפעם אחת).

מעניין כי לפעמים גם אחרי תיקון השגיאה המודל מופיע את התשובה כלל נכון או "לא יודע האם נכון". המחברים גם מציעים שיטה נחמדה לפתור בעיות מתמטיות על ידי הרצות חזרות של מודל שפה לאותה הבעיה המתמטית. לאחר מכן מחשבים ציון של כל תשובה על ידי משקל של תשובות כאשר כל תשובה שקיבלה "נכון" מקבלים ציון הגבוה ביותר, "לא יודע" מקבלת פחות ו"לא נכון" מקבל משקל 0. המחברים מוכחים שטריק פשוט זה מעלה את הסיכוי לקבל תשובה נכונה ממודל שפה.

**Question**

**Question:**  
What is the smallest positive integer solution to  $(\frac{100,000,000}{x^4}) < 1$ ?

**Answer:** 101

**(b) Example: w/ self verifications**

We can firstly change the inequality into  $x > \sqrt[4]{100000000}$ . Then let's evaluate the fourth root of 100,000,000.

```
code
import math
root = math.pow(100000000, 1/4)
smallest_integer = math.ceil(root)
>>> (100.0, 100)
```

Thus, the smallest positive integer  $x$  is 100. Let's verify this solution.

```
code
verification = 1e8 / (smallest_integer ** 4) < 1
>>> False
```

The smallest integer  $x$  should be the next integer greater than 100, not 100 itself. Let's calculate that next integer.

```
code
smallest_integer = 100 + 1
verification = 1e8 / (smallest_integer ** 4) < 1
>>> (101, True)
```

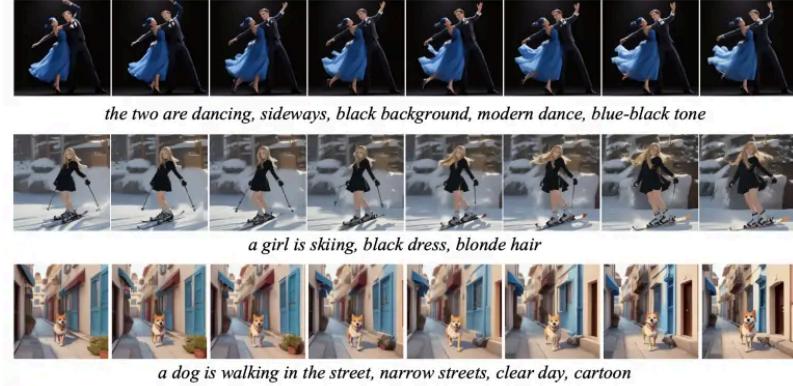
Upon substituting  $x = 101$  back, it holds true, which verifies our solution.

The final answer is 101 ✓

Figure 3: Question from the 712th intermediate algebra problem in the MATH dataset. (a) Without self-verification, the model generates a wrong answer. (b) With self-verification, the model corrects the error and generates the correct answer. The CSV prompt: *To solve the problem using code interpreter step by step, and please verify your answer using code interpreter.*

## Review 127, Short: Dual-Stream Diffusion Net for Text-to-Video Generation, 17.08.23

<https://huggingface.co/papers/2308.08316>



הגישות הראשונות ליצירה של וידאו מטקסט באמצעות מודלי דיפוזיה יצאו לראשונה לפני שנה וחצי ומazel השתדרגו משמעותית מבחינת איכות הוידאו, אורכו והתאמתו לתיאור. היום ניתן לעשות גם פרטוניליזציה למודלים אלו כולם ליצור וידאו עם אובייקט ספציפי (החתול שלכם).

המאמר שנסקור היום #-shorthereviewspaperreviews מshedרג את הגישה זו ומאפשר ליצור וידאו לא רק לאובייקט מסוים אלא גם לדפוס תנוצה מסוים (הנגזר מוידאו אחר למשל). וכל זה בהתאם לתיאור הטקסטואלי. איך באמנים מודל זהה? מזינים למודל את הוידאו ובשלב הראשון מעבירים כל פרימט דרך האנקודר להפקה של ייצוג הלטנטי (עם VQ-VAE).

מהיצוג זהה מאנשים מודל המפרק את הייצוג הזה את ייצוג התנוצה בוידאו (בין הפריים) לבין ייצוג התוכן של הוידאו (כל אחד מהם הוא מערכת של וקטורי ייצוג) – זה נעשה באמצעות MOTION DECOMPOSER. מערכת וקטורים

אלו מוזן למודל דיפוזיה משלו (מכאן בא השם dual stream) שעושים את קסמיים הרגילים. פלטי מודלי דיפוזיה אלו מוזנים לרשות ש"מערבתת" אותם ומוציאה שני ייצוגים מסונכרנים של תנועה ושל התוכן.

בסוף שני ייצוגים מסונכרנים אלו מוזן לרשות המשלבת אותם ובונה ייצוג של וידאו שעובר דרך הדקorder כדי לגנרט וידאו. כאשר רצים לגנרט וידאו לייצוג תנועה נתנו מכילים את המודול על ידי מזעור לוں השחזור את ייצוג התנועה מהייצוג היידאו המגנרט.

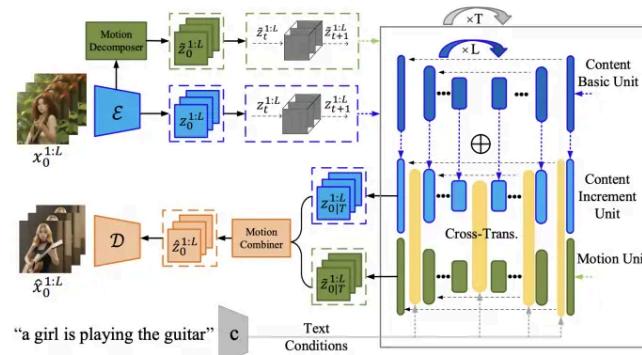


Figure 2: DSDN network framework. Initially, Content and motion features are added to noise during the diffusion process, followed by a denoising step via the dual-stream diffusion net. Lastly, the latent space features of the generated video are obtained through the motion combiner and decoded to render the final generated video.

## Review 128, Short: 18.08.23: CyBERT: Contextualized Embeddings for the Cybersecurity Domain (סקירה זו נכתבת על ידי עדן יבין)

<https://mdsoar.org/bitstream/handle/11603/25498/1117.pdf>

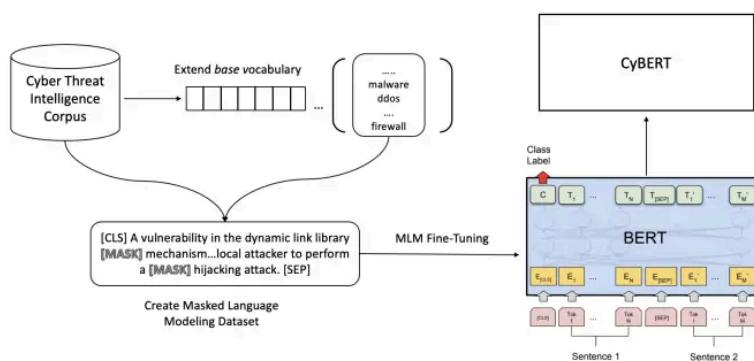


Fig. 1. Architecture Diagram for fine-tuning of the pre-trained BERT model to create CyBERT.

מודלי שפה הראו את היכולת שלהם לעבוד בהמוני תחומים בתוך עיבוד שפה טבעי. אחד התחומים שבhem ציפו להשפעה גדולה של מודלים אלו הוא תחום אבטחת המידע או בשמו היותר מוקר תחום הסיביר. במאמר ששמו הינו CyBERT הראו לראשונה את השילוב של מודלי שפה גדולים (מודל BERT) בתחום הסיביר.

החוקרים ניסו להראות איך שיפור של BERT הקים יכול להביא לשיפור ניכר של אותו מודל בנסיבות שונות כגון

זהוי אובייקטים הקשורים לתמונה הסיבר או סיווג של מילה לאובייקט המתאים לה. למשל, ניתן לראות תוצאות של המשימה الأخيرة המזכרת בה המודל נאלץ לסיווג מילה מתוך המונחים הסיבר לאובייקט המתאים לה. איך ביצעו זאת? על ידי הרחבת האימון של BERT עם מספר שלבים נוספים:

- אוספים מסמכים רבים מתוך הסיבר
- מנוקים את המסמכים והופכים אותם לרשימה של טוקנים
- את הרשימה של טוקנים נוספים למילון של ה-Tokenizer של BERT. בנוסף, מוסיפים למטריצת Masked Language Embedding הרגילה של BERT את הטוקנים עם ערך רנדומלי. מבצעים אימון נוסף של סיבר במשפט. המאמר מטען אותו במיוחד לא רק בשל היותו בתחום הסיבר אלא בשל התזכורת שהוא נתן כיצד ניתן לבצע אימון נוסף של מודל שפה גדולים על תחומים חדשים.

## Review 129, Short: Watch Your Steps: Local Image and Scene Editing by Text Instructions, 26.08.23

<https://arxiv.org/abs/2211.09800.pdf>

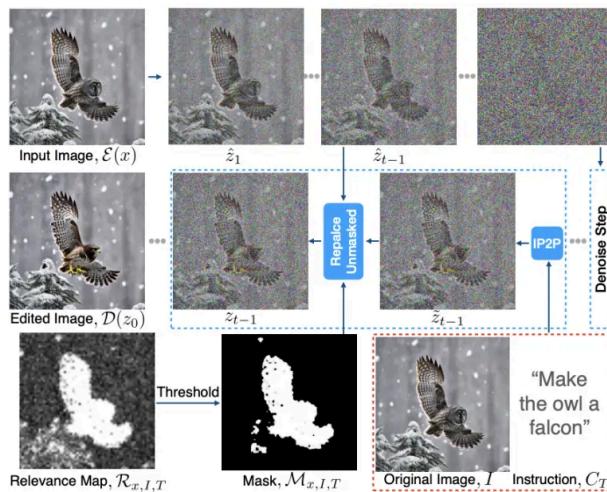


Figure 2. Overview of a denoising step for image editing via relevance-guidance. The relevance map is binarized to get the edit mask. After denoising the output of the last stage with IP2P, the unmasked pixels are swapped with the noisy input to ensure consistency to the input throughout the process.

מודלי הדיפוזיה לגמרי השתלו כמעט על כל המשימות של הראייה הממוחשבת. למשל עריכת של תמונות (למשל להחליף ציפור בפרפר) בהתאם לתיאור טקסטואלי כבר מזמן עושים רק באמצעות מודלי דיפוזיה חזקים כמו InstructPix2Pix או IP2P בקצרה. לעומת זאת התוצאות המדහימות עדין יש אי התאמות בין התמונה הערכוה לבין המקורית.

היום ב-#shorthebrewpapereviews נסקור מאמר שמנסה לתקן את אי דיוקים אלו בצורה די אלגנטית. בשלב

הראשון המודל המוצע מאתר את מיקום הפיקסלים שאותם צריך לשנות(מסכה) ובשלב השני עורכים את התמונה רק באזוריים של המסכה. כל זה נעשה באמצעות מודלי דיפוזיה באופן די אלגנט.

בשלב הראשון מראים את התמונה המקורי (עד רמת רעש מסוימת שהיא מהוות ה"ייפרפרמטר חשוב מאוד") משתמשים במודל IP2P כדי לשערר את הרעש נוסף עבור ללא תופסת טקסט לעריכה ויחד איתו. כולם במקורה הראשון אנו מפעילים מודל דיפוזיה סטנדרטי (ללא עריכה) ובמקורה השני כן עורכים את התמונה בהתאם לטיור הטקסטואלי. לאחר מכן מחשבים את הערך המוחלט של ההפרש בין השערוכים אלו, מקיצים את החירגים (עם הטקסטואלי). לאחר מכן מקבלים את הרעשים מושגים את הערך המוחלט של ההפרש הזה עולה על סף מסוים (ה"ייפרפרמטר נוסף").

בשלב השני מראים את התמונה (רמת הרעש עוד ה"ייפרפרמטר"). אז באמצעות מסירים את הרעשים עם מודל IP2P (עם טיור טקסטואלי) באזוריים של המסכה ובכל האזוריים האחרים עושים זאת עם מודל דיפוזיה רגיל (הטקסט שהוא ריק). בסופו המחברים מכלילים את הגישה שלהם ל-NeRF ("יצוג של מודלי 3D). בגודלousseים את מה שמתואר לעיל `saws` מכל חזיות תוך שמירה של קוהרנטיות בינויהם.



Figure 1. Overview of the calculation of the relevance map (left inset), and sample outputs on image (top-right inset) and neural radiance field (bottom-right inset) editing guided by the relevance. Given an image or a Neural Radiance Field (NeRF), our goal is to change the input according to a textual instruction. The relevance map is the disagreement between noise predictions with and without the instruction. For both image and scene editing, we use the relevance map to confine the changes to the most relevant region, according to the edit text.

## Review 130: SeamlessM4T—Massively Multilingual & Multimodal Machine Translation, 27.08.23

<https://ai.meta.com/research/publications/seamless-m4t>

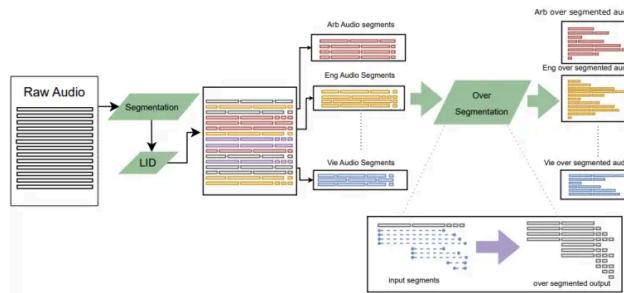


Figure 1: Workflow of speech processing.

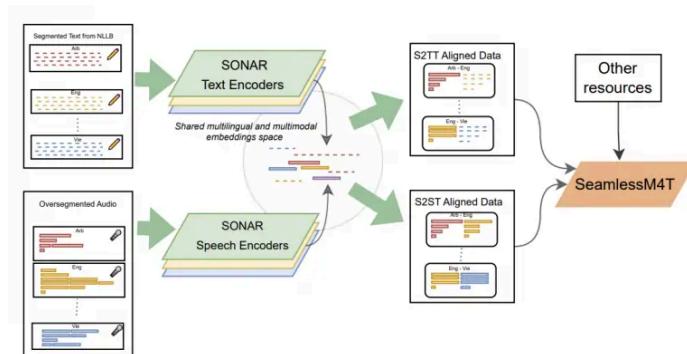
היום ב-#shorthethebrewpaperreviews נסקור מאמר מרשימים מבית היוצר של מטה שיצא לפני כמה ימים. המאמר מציע מודל מולטימודלי שתומך בשני סוגי של>Data בלבד: אותות קול (speech) ושפה. זה לא נשמע מרשימים מדי עד שבבאים מה המודל הזה מסוגל לעשות עם שני סוגי>Data האלו. למעשה המודל מאפשר לבצע 4 פעולות: תרגום של אות קול משפט מקורה לשפה אחרת, תרגום של אות קול לטקסט בשפה אחרת, הפיכה של אות קול לטקסט בשפה אחרת ותרגום טקסטואלי רגיל.

זה גם לא נשמע מרשימים במיוחד במקודם העמוסה במודלים עצמאים שיצאים כמעט כל יום עד שמגלים ה- SeamlessM4T יודע לבצע את הפעולות האלו ללא פחות מ-200 שפות שונות כולל שפות דיאלקטיות כמו אורהגית ולטאית. כמו שכבר אמרנו בסקרים הקודמות הדבר חשוב ביותר במודלי מילויים מודלים הוא מיפוי קורגרנטי של סוג הדאטה השונים לאוטו מרחב שיכון (embedding space).).

הקוורנטיניות כאן פירושה שפיזיות דатаה מסווגים עם משמעות דומה יהיו קרובים למרחב זהה ואלו שימושם שונים יהיו רוחקים זה מזה. מכיוון שהמבנה הפנימי של אות קול הוא מאוד שונה מטיקסט הוא צריך לעבור עיבוד מוקדים (preprocessing) לפני שמקניתים אותו לרשף המיפה אותו למרחב זהה.

מכיוון שגלי קול בשפות שונות מאוד מבנה הראשוני מזהים את השפה (כאשר הקלט מכיל כמה שפות מפרקם אותו לSEGMENTATION שכל אחד הוא בשפה שונה). לאחר מכן לוקחים אותן דיבור בכל שפה ועושים מה שנקרא OVERSEGMENTATION לאגוזדר של אותות קול שמאמנו להפיך וקבעו שיוכן של אות קול.

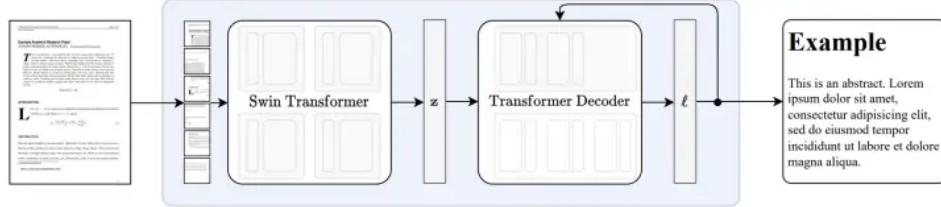
המאמר גם מאמין אנקודור לטקסט בצורה די סטנדרטיבית (אך לשפות רבותות). לאחר מכן מאמנים מודל המיפה שיכונים אלו למרחב יצוג משותף ולאחר מכן מאמנים דקודור ההופך את יצוג זהה לטקסט. בשלב האחרון מאמנים מודל נוסף ההופך את הטקסט הזה לאות קול. יש הרבה פרטים מעניינים על תהליך האימון שלא ניתן לכנותם בסקירה הקצרה זו – ממליצ להעיף מבט.



**Figure 2:** Workflow of the SONAR encoding and mining processes.

**Review 131, Short: 28.08.23: Nougat: Neural Optical  
Understanding for Academic Documents**

<https://arxiv.org/abs/2308.13418.pdf>



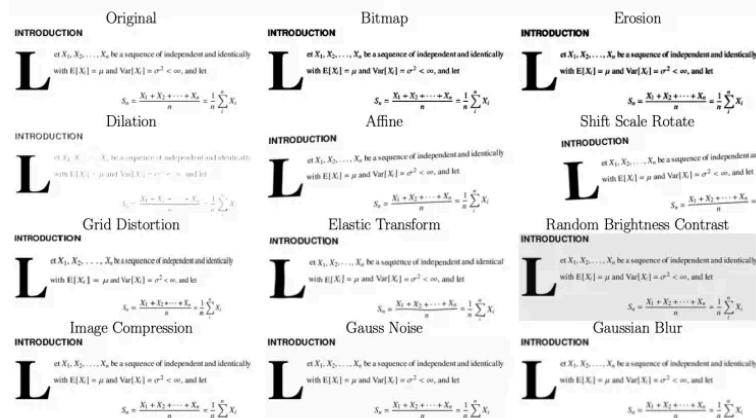
**Figure 1:** Our simple end-to-end architecture followin Donut [28]. The Swin Transformer encoder takes a document image and converts it into latent embeddings, which are subsequently converted to a sequence of tokens in a auto-regressive manner

אתם בטח שמתם לב כי אני אוהב לקרוא מאמרים. יש מאמרים שלא פשוט להבין אותם ואז אני מתחילה לחפש מושגים שונים או איפה ובאיזה הקשר מופיעים כל מיני REFERENCES בקובץ של המאמר. זה כמובן אפשרי רק עבור מאמרים ייחשים חדשים (ב 20 השנים האחרונות). במקרים יישנים זה בעיתוי כי הם פשוט מהווים צילום של המאמר.

היום ב-#shorthbrewpapereviews סוקרים מאמר שלוקח מאמר מדעי שהוא לא בפורמט הננו והופך אותו למסמך נוח לקרוא באיכות גבוהה ולחפש שם כל מיני דברים. איך הם עושים זאת? מכיוון שהמסמך מגיע בתור תמונה בשלב הראשון צריך לבצע OCR Optical Character Recognition או OCR. משימה זאת אינה פשוטה כי מאמרים לפעמים מכילים נוסחים דומים וסביר להניח שישות OCR קיימות די יתקשו להתמודד עם זה.

המאמר מפתח מודל לזהוי תוכן מהתמונה של מסמך עצמו. המודל מורכב מאנקודר שמקבל את תמונה המאמר, עושה לו עיבוד מקדים (מוריד שלויים, הופך אותו גדול וכדומה). לאחר מכן מכון התמונה מחולקת לפאות זרים ומכניסה אותו לרשות הטרנספורמר מסווג Swin שמטרתו להפיק את הייצוג הלטני של תמונה המסמך. לאחר מכן ייצוג לטני זה מזון לדקORDER שהוא גם טרנספורמר שמטרתו לענча את המסמך ולהציג אותו בשפת markdown, שניתן להפוך אותה ל-pdf בקלות.

הדאטהסט לאימון בני מתנות של מאמרים ומייצגים בשפת markdown (למסמך קיימים בצורה "דיגיטלית" ניתן לתרגם את ייצוג-h-tex של המסמך לשפת markdown הזו). כמובן משתמשים במגוון אוגמנציות של תונות המסמכים לאימון המודל שלהם כדי לשפר את יכולת ההכללה שלהם. כמובן השיטה המועצת עדין מוגבלת ודורשת עיבוד מקדים לא קל של תונות המאמרים אבל זו התחלת טוביה.



**Figure 2:** List of the different image augmentation methods used during training on an example snippet form a sample document.

# Review 132: Beyond Chain-of-Thought, Effective Graph-of-Thought Reasoning in Large Language Models, 29.08.23

<https://arxiv.org/abs/2305.16582.pdf>

הסירה נכתבת על ידי עדן בון

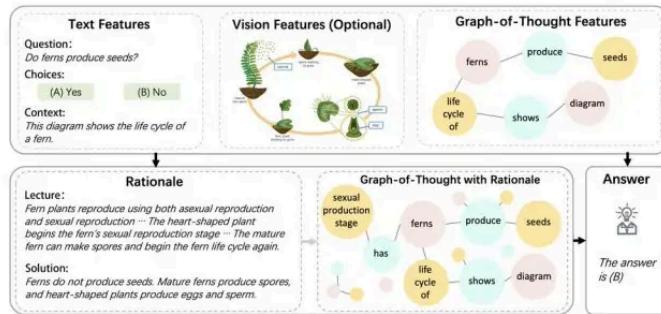


Figure 1: An example of GoT reasoning. Vision features are optional and are only required in multimodal reasoning task.

מנגנוני היגיון שונים עוזרים לסייעו להבין מושגים מורכבות ולבצע אותן בהצלחה. המנגנונים כוללים שיטות כמו שרשרת מחשבות (chain of thought) או עץ מחשבות ארכ מה עם גרפ' מחשבות?

היום ב-[shorthebrewpapereviews#-](#). החוקרים טוענים שהדרך שלהם מבוססת על העבודה שנעשים לא חשבים בצעדים כ皋ן בשרשראת אלא קופצים "מקודקוד" של מחשבה לקודקוד אחר. כך ניתן להרכיב מחשבות מורכבות ומגוונות יותר על ידי חיבור קשרים שונים בגרף. אבל איך זה מתבצע? בואו נגלה.

1. בניית הגרף מהטקסט – בהינתן הטקסט, החוקרים מחליצים ממנו שלישיות של נושא-פועל-נושא (למשל רעיתת אדמה מגעה מרuida ואדמה). מהשלישיות הם מפעלים אלגוריתם אשכולות Extract-Clustering-Coreference למשול רעיתת אדמה-מגעה-רעידה ואדמה | רעדיה, אדמה-שהמשמעות שלהם-אדמה ורואה. لكن זה יהיה אשכול שמננו ניתן להסיק שרuidת אדמה -> אדמה, רועדת.

2. הגרף מקודד באמצעות Graph Encoder המבצע שיכון של כל קודקוד על ידי שימוש בשכנים שלו (כלומר משתמשים במטריצת השכניםות של הגרף).

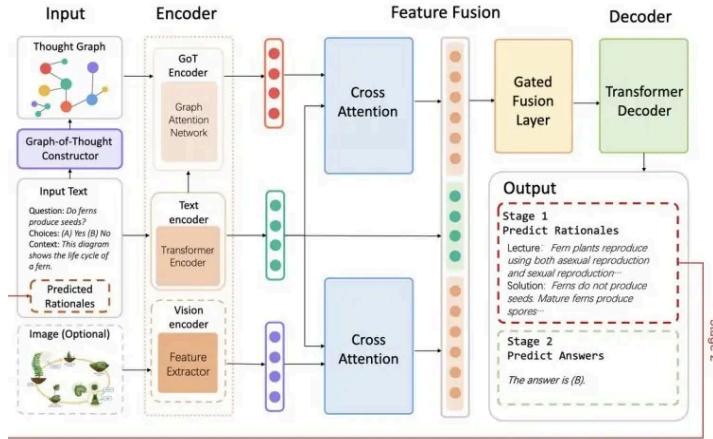
3. הטקסט מקודד באמצעות Encoder רגיל

4. תמונות (אם קיימות) מקודדות באמצעות Vision Encoder.

5. משתמשים ב- cross attention כדי לבצע תיקון לקידודים כך שהשיכון יהיה באותו מרחב קידוד.

6. מאחדים את כל הקידודים ביחד ומשתמשים ב- Transformer Decoder בשביל לייצר את התשובה

לצורך אימון הם אימנו על הדאטאסתים: Multimodal ScienceQA - ו- text-only GSM8K. התוצאות מראות שיפור מול המתחרה Co شמשתמשים במודל זהה בגודל למודל שלהם.



## Review 133, Short: OmniQuant: Omnidirectionally Calibrated Quantization for Large Language Models, 30.08.23

<https://arxiv.org/abs/2308.13137>

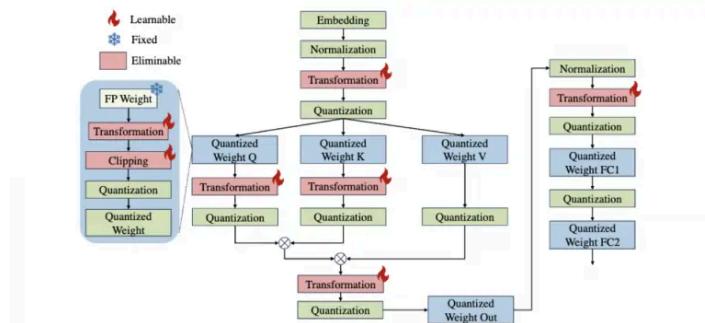


Figure 3: **Details of OmniQuant** in a transformer block. Note that the learnable equivalent transformation can be absorbed by the normalization layer and linear layer. The learnable clipping threshold applies only to float-point weight and thus can be discarded after quantization. Therefore, OmniQuant does not introduce any additional computation cost or parameters after quantization.

קונטיזציה (quantization) של מודלי שפה הפך להיות נושא חם לאחרונה. מודלי שפה הפכו להיות ענקיים ומכלולים עשרות מיליארדי פרמטרים וגודלם נמדד בגיגבייטים. כדי לאפשר הפעלה מהירה של מודלי שפה (וגם חסכון במקום האחסון).

היום ב-#shorthereviewpaperreviews סוקרים גישה חדשה לקונטיזציה של מודלי שפה המשלבת טכניקות שהוצעו בכמה עבודות קודמות וגם מציעה כמה חידושים. נתחיל מרגען מהי מטרת הקונטיזציה? המטרה היא "לדוחוו" את מודל השפה באופן שהפגיעה בביטויים (inference) תהיה מינימלית (בד"כ בהינתן תקציב דחוסה נתנו לנו מועד מוקדם הדחיסה).

קודם כל המחברים מיצגים את בעיית הקוונטיזציה בבעית מינימיזציה (minimization) של הפרש בין תוצאה החישוב של המודל ב-precision מלא והותואה של המודל המוקוונט (על סט ולידציה) עבור כל בלוק טרנספורמר (יחידה בסיסית של כל  $\text{LMS}^2$  היום). עכשו נשאלת השאלה מה הפרמטרים של בעית מינימיזציה?

המאמר מציע לשלב שתי טרנספורמציות נלמדות (פר בלוק הטרנספורמר): הראשונה היא פועלות קוונטיזציה עצמה של משקל המודל (מתבצעת באמצעות טרנספורמציה max-min עם שני פרמטרים של scaling נלמדים).

הפעולה השנייה היא טרנספורמציה לינארית של פלטי הבינים של המודל עם פרמטרים נלמדים. למשל הפלט של בלוק טרנספורמר מסוים מכפל במטריצה ומבודד (פרמטרים נלמדים) לפני שהוא נכנס לבלוק הטרנספורמר הבא. נציין כי יש פרמטרים שונים לטרנספורמציה לינארית בכניסה למנגנון תשומת הלב. שילוב קווניטוט עם טרנספורמציה של הפלט (משום מה נקרא *activation* במאמר שזה קצת מבלבל) מביא לדחיסה יعلاה של מודל שפה עם פגעה מינורית בביטויים.

**Algorithm 1** Overall algorithm of OmniQuant.

**Input:** calibration dataset  $\mathbf{X}$ , pre-trained LLM model  $\mathcal{M}$

**Output:** quantized model.

```

1:  $\mathbf{X}_{fp} = \mathbf{X}_q = \mathbf{X}$                                 ▷ init inputs of full-precision and quantized models.
2: for  $\mathcal{B}_i$  in  $\mathcal{M}$  do:                               ▷ block-wise calibration
3:    $\mathbf{X}_{fp} = \mathcal{B}_i(\mathbf{X}_{fp})$                   ▷ update the input of full-precision model
4:   init learnable weight clipping parameters  $\Theta_1$ 
5:   init learnable equivalent transformation  $\Theta_2$ 
6:   for k in epochs do:
7:     for  $(\mathbf{x}_q, \mathbf{x}_{fp})$  in  $(\mathbf{X}_q, \mathbf{X}_{fp})$  do
8:        $\mathcal{B}'_i = \text{LET}(\mathcal{B}_i, \Theta_2)$            ▷ With Eq.(3),Eq.(5)
9:        $\mathcal{B}'_i = \text{Quantization\_with\_LWC}(\mathcal{B}'_i, \Theta_1)$     ▷ With Eq.(2)
10:       $\mathbf{x}'_q = \mathcal{B}'_i(\mathbf{x}_q)$ 
11:      loss =  $\|\mathbf{x}_{fp} - \mathbf{x}'_q\|^2$                 ▷ With Eq.(1)
12:      loss.backward()
13:      update  $\Theta_1$  and  $\Theta_2$  through gradient
14:    end for
15:   end for
16:    $\mathcal{B}_i = \text{LET}(\mathcal{B}_i, \Theta_2)$ 
17:    $\mathcal{B}_i = \text{Quantization\_with\_LWC}(\mathcal{B}_i, \Theta_1)$           ▷ obtain the quantized block
18:    $\mathbf{X}_q = \mathcal{B}_i(\mathbf{X}_q)$                            ▷ update the input of quantized model
19: end for
20: return quantized model  $\mathcal{M}$ 

```

## Review 134: LM-INFINITE: SIMPLE ON-THE-FLY LENGTH GENERALIZATION FOR LARGE LANGUAGE MODELS, 31.08.23

<https://arxiv.org/abs/2308.16137.pdf>

אורק ההקשר (context length) של מודלי שפה או במילים אחרות הגודל המקורי של הטקסט המודל שפה יכול "לזכור" הפק להיות נושא מאוד פופולרי בקהלית-NLP עקב חшибתו הרבה ליישומים רבים. יצאו عشرות (אם לא מאות מאמרים) המנסים להגדיל אותו והיום ב-shorthereviewspaperstheseemsano נסקור מאמר המציג גישה חדשה להערכת אורק הקשר.

קודם כל המחברים מזהים (וסוג של מוכחים) למה מודלי שפה שאומנו עם קלט קצר יחסית ועם קידוד תיל מיקום יחסיים (RoPE) מתקשים עם קלט יותר ארוך באינפראנס. אז לפי המאמר יש 3 סיבות עיקריות.

- או ש- rope מתעלם מהתוקנים הרחוקים (מקדמי attention לפני softmax שוויים ל-0) או שהם מקבלים ערכיהם גבוהים מאוד
- כל שمارיכים את אורך ההקשר האנתרופופיה של מקדמי-h attention שואפת ל-N(log) כאשר N זה מספר התוקנים כלומר המודל מתחשב בכל התוקנים באותו מידה
- המודל מקודד באופן לא מפורש את המיקום האבסולוטי של התוקנים בסדרה (המאמר מסביר את זה בkr ש"הסיגן מהתוקנים ההתחלתיים חזק יותר מהאליה שבסוף")

כדי להתמודד עם 3 סוגיות האלה המאמר מציע גישה די פשוטה לקידוד תלוי מיקום:

- עבור טוקן נתון הם מקודדים (עם PE RoPE או משחו דומה) מספר טוקנים מסוימים (גדי שווה לאורך ההקשר "הסטנדרטי" של מודל שפה) המופיעים **בתחילת הסדרה** (global branch)
- כל טוקן גם מקודד (attend) את מיקום טוקנים הנמצאים במרקח מסוים ממנו (local branch)

לפי המאמר כך התוקנים שבהתחלתה בעיקר מקודדים בעיקר את המיקום האבסולוטי של הטוקן, אלה שבסוף את המיקום היחסי שלו ואלו שבאמצע מכילים פחות מידע מיקומי (לא הבנתי את למה בעצם).

## Review 135, Short: ORES: Open-vocabulary Responsible Visual Synthesis, 01.09.23

<https://arxiv.org/abs/2308.13785.pdf>

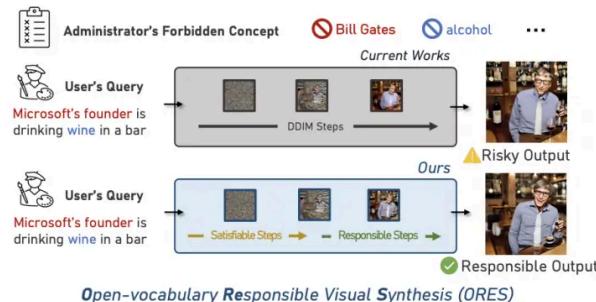


Figure 1: **Open-vocabulary responsible visual synthesis.**  
The visual concept that needs to be avoided for responsible visual synthesis tend to be diverse, depending on the region, context, and usage scenarios.

כיום מודלים לגנרטו תמונות לפי תיאור טקסטואלי מסווגים ליצור תמונות באיכות מדיה שימוש תואמת את התיאור. בנוסף נהיה קשה מאד עד בלתי אפשרי לבדוק אם תמונה הינה "טבעית" נוצרה על ידי מודל גנרטיבי שפותח פותח ליצור פיקים באיכות גבוהה מאוד שעולים לנזק רב.

היום ב-#shorthebrewpapereviews סוקרים מאמר שפיתח שיטה פשוטה למניעת יצירת פיקים מסווגים. השיטה המוצעת מאוד פשוטה אינטואיטיבית. נניח שיש לנו פרופט שבאצעתו המשתמש רוצה ליצור תמונה. בנוסף יש לנו גם ייש לנו גם סט של קונספטים (מושגים) אסורים (כמו ערום, דם וכאלו). בשלב הראשון הופכים את הפרופט לבטווח (בהתאם לקונספטים האסורים) עם מודל שפה מואום.

המחברים בחרו דאטסהט קטן המכיל שלישיות של (פרומפט, מושג אסור, פרומפט בטוח) ובהתבסס עליו מצאו את הפרומפט (אחד) למודל שפה שבאמצעותו ניתן להפוך הנחיה נתונה יחד עם הקונסupt האסור להנחיה בטוחה על ידי הרצה של הדאטסהט זהה על מודל שפה מסויף אפוקים. לאחר מכן מכנים את הפרומפט בטוח למודל דיפוזיה מאומן. כדי לגנרט תמונה דומה לפромפט המקורי ב- S הצעדים ההתחלתיים של מודל הדיפוזיה (מתחלים מרعش טהור) וב- S – T הצעדים האחרונים מבצעים עם הפרומפט הבטווח.

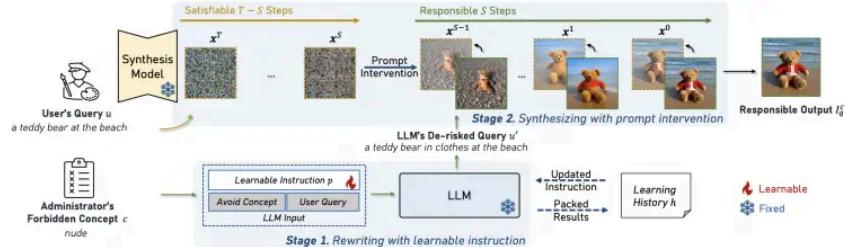


Figure 2: **Overview of the TIN framework.** TIN can be divided into two stages: (1) rewriting with learnable instruction, and (2) synthesizing with prompt intervention.

## Review 136, Short: Any-Size-Diffusion: Toward Efficient Text-Driven Synthesis for Any-Size HD Images, 03.09.2023

<https://arxiv.org/abs/2308.16582.pdf>

A cute teddy bear in front of a plain white wall. The teddy bear has a warm, brown fur that looks soft and fluffy, sitting on the brown wooden tabletop.



Figure 1: **Resolution-induced Poor Composition.** Given the text, (a)  $SD_{2,1}$  and (b)  $MD_{2,1}$ , a MultiDiffusion model, raise poor composition issues in red boxes when synthesizing images of varying sizes, as opposed to (c) our ASD.

אוק", אני חייב להתודות בפניכם. למרות שעסוקתי לא מעט במודלי דיפוזיה גנרטיביים לא ידעתם שהם מתקשים ליצור תמונות בגודלים שונים. אז היום גיליתי את זה דרך המאמר שאני הולך לסקור קצרות ב-#shorts#thebrewpapereviews.

המאמר מציע גישה די פשוטה לאימון מודלים לגודלים שונים. קודם כל לוקחים נתונים עם תמונות (בעלות כותרת או תיאור) מכל מיני גודלים, ועבור כל תמונה מחשבים יחס' של אורך לרוחב (W/H). מגדרים את היחסים

בין רוחב לגודל שהמודול מאמין עליהם ומחפשים את היחס הקרוב ביותר לזה של התמונה.

از עושים לתמונה סקילינג לפי היחס הזה ומכניםים אותה למודל דיפוזיה מאמין (Stable Diffusion-SD). לאחר מכן מכילים את המודל זהה (כמו שמאנים מודל דיפוזיה רגיל) אבל בשיטת LoRA כאשר האנקודר והדקודר לא מתעדכנים (frozen). ככלומר מאנים מטריצת תוספות למטריצות משקלים של המודול המאמין (שנשאר מוקפאת) כאשר מטריצת תוספות זו מאומנת בתור מכפלה של שתי מטריצות low-rank (חיזור הכלול).

איך מגנרטים תמונות לאחר מכן? יש להם שיטה מעניינת (הופעתה קצר שהיא עבדת) – מחלקים תמונה לפאות זרים ואז בכל איטרציה של מודל דיפוזיה בוחרים כמה מהם, מזינים אותם לרוחב ולאורך עם היחס המקורי, משחררים את הרעש הנוסף בפאצ'ים המוזדים ואת השאר מעתיקים מהאיטרציה הקודמת. כאמור מפטיע שזה עובד כי זה קצר לא מתיישר עם אופן האימון של מודל דיפוזיה – נראית מספר האיטרציות הוא די גבוה כאן.

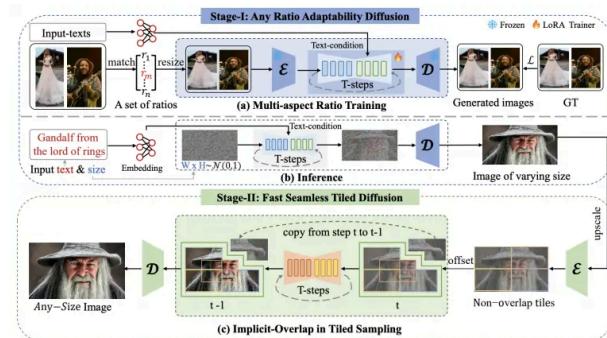


Figure 2: The Any-Size-Diffusion (ASD) pipeline, including: 1) Stage-I, Any Ratio Adaptability Diffusion, translates text into images, adapting to various aspect ratios, and 2) is responsible for transforming low-resolution images from the Stage-I into high-resolution versions of any specified size. For procedure (c), the implicit overlap in tiled sampling, only the solid green line region is sent to the UNetModel for current denoising. At Stage-II, the dashed green arrow represents regions that are directly copied from the preceding denoised latents, potentially enhancing efficiency and consistency within the overall process.

## Review 137: YaRN: Efficient Context Window Extension of Large Language Models, 04.09.2023

<https://huggingface.co/papers/2309.00071>

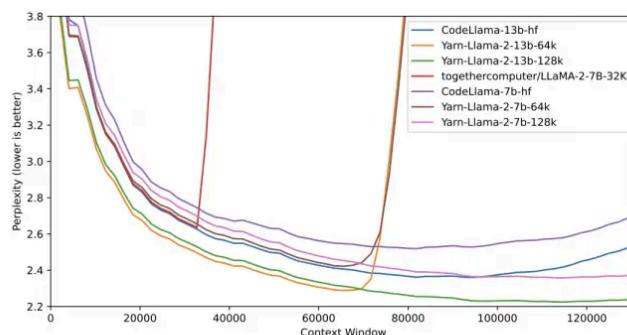


Figure 1: Sliding window perplexity ( $S = 256$ ) of ten 128k Proof-pile documents truncated to evaluation context window size

מכירים את בעיית הרחבת אורך ההקשר (CL=context length) של מודלי שפה? בגודל הסוגיה כאן אין להתאים מודל שפה לעבוד עם טקסטים ארוכים יותר מאשר אומן עליהם. בהקשר זה קיימות 2 בעיות שונות: איך לגרום למודל שפה לעבוד עם CL ארוך יותר ואיך לכילו ל-CL ארוך יותר?

از היום ב-shorthereviewspaperreviews סוקרים מאמר המציג כמה גישות לפתורן של סוגיה זו. המאמר מציע לאתגר לפתרון דרך התאמת של קידוד תלוי מיקום (PE = positional encoding) שבא לקודד מיקום של טוקנים בסדרת הפלט. המאמר מציע לשככל שיטת PE הנקרא RoPE שמקודדת את מיקום התוקן יחד עם ייצוג הטוקן (embedding) כאשר כל מיקום מקודד עם וקטור של רכיביו הוא אקספוננט מרוכב (עם ? בתוכו) מוכפל (אייר) עם רכיבי וקטור ייצוג הטוקן עצמו.

כלומר לכל רכיב של וקטור ייצוג התוקן מוספים פaza (הזה) שהוא תלוי במיקום התוקן בסדרה וגם במיקום הרכיב בוקטור הייצוג. קודם כל המחרבים שמו לב שבעיתות ייצוג המיקום קשורה לתורת NTK (Neural Tangent Kernel). תורה זו אומرت כי רשותות נוירונים מתקשות לקודד מידע בעל תדרים גבוהים (כמו קצנות מובחרים בתמונות) עם הייצוג הקלט הוא בעל מימד נמוך יחסית וייצגו לא מכיל תדרים גבוהים. במודל שפה אנו מננים לקודד את מיקום התוקן בסדרה עם וקטור מרוכב(סינוסים וקוסינוסים).

המאמר מדגיש שהגישה הטריאלית להרחבת חלון ההקשר על ידי הכפלה של כל התדרים של וקטור הייצוג ביחס בין חלון ההקשר החדש לשין (היחס הזה גדול מ 1) גורמת להפסד של תדרים גבוהים שהרשות צריכה בשילוב לבחון בין טוקנים קרובים בעלי ייצוגים קרובים.

בגודל (מאוד) המאמר מציע לא למתוח את כל התדרים במידה שווה (הכפלה ביחס בין אורכי ההקשר) אלא לעשות זאת רק לתדרים נמוכים יחסית (טוקנים ראשונים) ולא לעשות זאת (הכפלה) לטוקנים בסוף הסדרה כאשר טוקנים באמצעות נמחחים גם כן אבל עם מקדם יותר קטן מהיחס הזה. כਮון שההמש בגודל, מציע בחום לרקוא את המאמר הזה כי אני מרגיש שהוא עוד יעשה כותרות.

Model Size	Model Name	Context Window	Extension Method	8192	32768	65536	98304	131072
7B	Together	32k	PI	3.50	2.64	$> 10^2$	$> 10^3$	$> 10^4$
7B	Code Llama	100k	NTK	3.71	2.74	2.55	2.54	2.71
7B	YaRN ( $s = 16$ )	64k	YaRN	3.51	2.65	2.42	$> 10^1$	$> 10^1$
7B	YaRN ( $s = 32$ )	128k	YaRN	3.56	2.70	2.45	2.36	2.37
13B	Code Llama	100k	NTK	3.54	2.63	2.41	2.37	2.54
13B	YaRN ( $s = 16$ )	64k	YaRN	3.25	2.50	2.29	$> 10^1$	$> 10^1$
13B	YaRN ( $s = 32$ )	128k	YaRN	3.29	2.53	2.31	2.23	2.24

Table 1: Sliding window perplexity ( $S = 256$ ) of ten 128k Proof-pile documents truncated to evaluation context window size

## Review 138: G-EVAL: NLG Evaluation using GPT-4 with Better Human Alignment, 05.09.2023

<https://github.com/nlpyang/geval/tree/main>, <https://arxiv.org/abs/2303.16634.pdf>

סקירה זו נכתבת על ידי עדן יבין

תהליך אבלואטיה של מודל עוסק בהערכת הביצועים שלו. אבל איך נוכל להעריך טקסט שנוצר על ידי מודל גנרטיבי? מדריכים שונים כגון BLEU, G-EVAL, מאפשרים לבצע הערכתה אף צורכים טקסט נוסף שנכתב על ידי אדם כרפרנו. בעיה, כי השגת טקסט שכזה הינו יקר, מה הפתרון אז?

המאמר נכתב על ידי חוקרים ממיקרוסופט ומציג שיטה הנקראת G-EVAL. השיטה מבוססת משתמש ב-GPT כדי להעריך את ביצועי מודל השפה שיצר טקסט כלשהו בהינתן איזו שהוא משימה. היא מדמה תהליך של מיילוי שאלון הערכה בכך שמאפשרת להגדיר קритריונים שונים חשובים לנו, החוקרים, להערכת איכות הטקסט. אין השיטה עובדת:

1. המשמש יוצר prompt בו הוא מתאר מה המשימה שמודל אחר ביצע + הסבר על הקритריונים להערכת הטקסט.
2. שימוש במודל שפה נוסף כדי לפרק את שיטת האבלואטיה לכמה עדינים, למשל: "תקרה את כל הטקסט ושים לב לכמה מילות הקישור", "תקרה כל משפט ושים לב לכמה המילים במשפט" וכדומה. שלב זה מבוצע על ידי שימוש ב-CoT (Chain of Thought) כדי לייצר את השלבים השונים האלו המדמים טופס הערכה עם נתוני סעיפים.
3. נתונים למודל השפה לתת את הצינונים השונים לכל קритריון שהגדכנו בשלב אחד לטקסט שנרצה להעריך. הציון צריך להיות מספר שלם ממשום שניסויים שנערכו מראים שהם מתקדים עם float.
4. מגדירים פונקציית ניקוד אשר משקללת גם את הניקוד יחד עם התפלגות הניקוד מהתפלגות הטוקנים של המודל. הסיבה היא למקורה ויהיה שווין בין הניקוד של טקסטים שונים נוכל להתחשב בניקוד יחד עם שקלול ההסתברות לקבל את אותו ניקוד.

כדי להשיג את התפלגות של הטוקנים מ-GPT4 הם דגמו כל קритריון 20 פעמים כדי לקבל את התפלגות הטוקנים והשתמשו בה בפונקציית הניקוד (יכול להיות ש-20 לא מספיק לעניות דעתך). עם GPT3.5 ניתן לשירות להציג את ההסתברות לכל טוקן אך אין צורך לבצע זאת עם מודל זה. את התוצאות הם השוו מול מדריכים בתחום. התוצאות כולן הן מול הערכה שביצע אדם על הטקסט לפי הקритריונים שהוגדרו מראש (אומנם קритריונים שהשתמש גם במודל). ניתן לראות שהשיטה של החוקרים הביאה לתוצאות טובות באופן ניכר יותר קרובות להערכתה שבוצעה על ידי אדם.

# Review 139: Unsupervised Compositional Concepts Discovery with Text-to-Image Generative Models, 06.09.2023

<https://arxiv.org/abs/2306.05357.pdf>

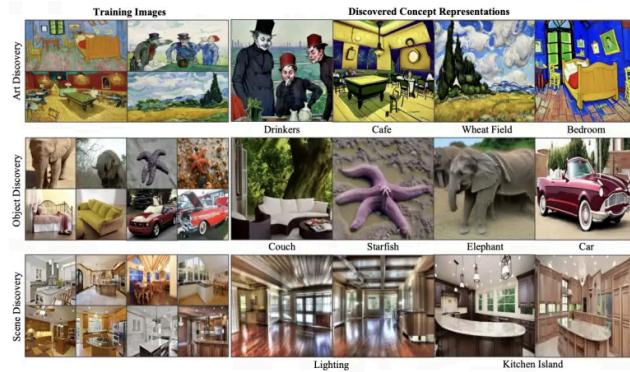


Figure 1: Unsupervised Image Decomposition. Our approach is able to decompose a dataset of unlabeled images into different concepts. We name each decomposed concept for easy understanding.

מודלי דיפוזיה גנרטיביים מסוגלים ליצור תמונות (ולא רק) באיכות מרתקה בהתאם לתיאור טקסטואלי. עכשו השאלה האם ניתן לגרום למודלי דיפוזיה לעשות את הפעולה ההפוכה (סוג של). קלומר להפיק קונספטים ויזואליים (כלב, מכונית וכאלו) מסט תמונות נתון.

היום ב-#shorthebrewpapereviews אנחנו סוקרים מאמר שמציע שיטה אלגנטית אינטואיטיבית לכך. קודם כל המאמר מגדיר מושג "מודל דיפוזיה מרכיב" (CDM composite diffusion model) או M. אתם זוכרים שבמהלך הගנות של תמונה אם מודלי דיפוזיה רגילים אנו מתחילה מרעש טהור ואז מסירים מנות קטנות של רעש, המוחש עם מודל מאומן, באיטרציות עד שמגיעים לתמונה נקייה.

המודל שמשערק את הרעש מקבל גם את התיאור של התמונה (יצוגו הוקטורי). עם M אנו יוצרים תמונה עם יותר מkoncept ויזואלי אחד כאשר הרעש המנוקה הוא סכום של שعروci רעש בהינתן koncept מינוס הרעש המשוער ללא התנינה בתיאור (unconditioned). זה די דומה (מקרה פרטי) של classifier-free guidance שיטה פופולרית לגנות תמונות עם מודלי דיפוזיה.

אבל בוואנו נחזור לשאלת המקורית: איך ניתן להפיק konceptים ויזואליים (או ייצוגם) מסט תמונות נתון? נניח שאנו רוצים "למוד" K konceptים ויזואליים מסט תמונות. אז כל תמונה ניתנת לייצוג בתור CDM כאשר במקום סכום של שعروci הרעש בהינתן כל koncept בוים את סכום המקומות (סכום המקדים שווה ל 1 וכלם אי שליליים) עם המקדים נלמדים עבור כל תמונה (גם כאן מחסירים מהסכום את שعروci הרעש unconditioned).

از בהינתן סט תמונות מכילים מודל דיפוזיה כאשר המטרה ללמידה את ייצוגי הקונספטים הויזואליים וגם המקדים של כל koncept בכל תמונה. הלוס מכך עד כמה טוב האלגוריתם לשחזר כל תמונה. לאחר מכן ניתן לגנרטת תמונה עבור כל koncept בנפרד או כל מיני שילובים שלהם.

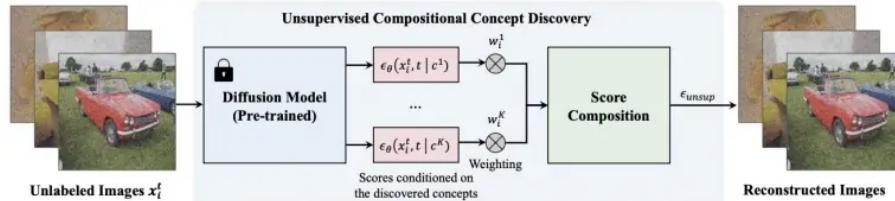


Figure 2: **Compositional Concept Discovery.** We discover a set of compositional concepts given a dataset of unlabeled images. Score functions representing each concept  $\{c^1, \dots, c^K\}$  are composed together to form a score function  $\epsilon_{\text{unsup}}$  that is trained to denoise images. The inferred concepts can be used to generate new images.

## Review 140, Short: One Wide Feedforward is All You Need, 07.09.2023

<https://huggingface.co/papers/2309.01826>

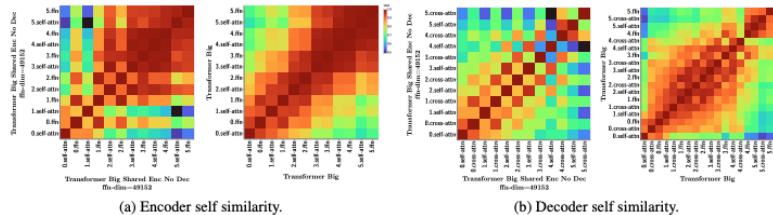


Figure 1: Self similarity structure of encoder and decoder layers of the *One Wide Encoder* model vs. the Transformer Big baseline.

ארQUITקטורת הטרנספורמרים היא המלכה הבלתי מעוררת של עולם AI. רוב המודלים כמו מודלי שפה או מודלי דיפוזיה גנרטיביים המככבים היום בחדשות AI בנויים על הארכיטקטורה הזאת. כמובן שיש לא מעט מחקר גם באקדמיה וגם בתעשייה על שיפור ביצועי הטרנספורמרים.

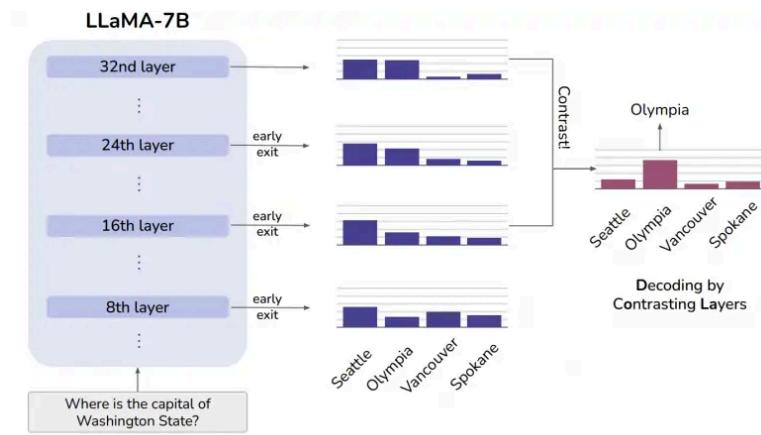
از היום ב-awsviews shorthebrewpaperereviews נסקור מאמר שמנסה לשפר שני היבטים של הטרנספורמרים: נפח האחסון וכמות חישובים (בכיוון הקטנותם). קודם כל נזכיר שככל בלוק של טרנספורמר (שהוא גרעין של כל מודול fully-connected על הטרנספורמרים) בני מנגנון של תשומת הלב (attention) ועוד שתי שכבות של attention עם ReLU והשניה לינארית.

לפי המאמר השכבות האלו מהוות 2/3 ממספר המשקלים (ב-BERT) וכמוון "טורמים" לעומת השכבות. המחברים שאליהם מה יקרה עם נזוטר על השכבות האלו או שנעשה אוטם "שיתופים" (shared) בין כל בלוקי הטרנספורמרים של המודול. זה עתיד להקטין את כמות המשקלים במודל באופן משמעותי כי רוב המודלים מכילים عشرות אלפי או מאות בלוקי הטרנספורמרים.

המאמר גם מציע "לשטי" משקלים בין האנקודר לדקודה. אבל איך זה משפייע על ביצועים. המאמר מראה שהפגעה ביצועים לא גדולה במיוחד במיחוד (למרות שהם ביצעו מספר בדיקות די מוצמצם והם בדקו זאת על מודלים די קטנים עם 6 בלוקי טרנספורמרים בלבד). הם גם השוו דמיון בין הייצוגים של משפטים עברו המודל המקורי והמודל "הקל" המוצע וגלו שהוא די גבוה. בנוסף הם השווו 2 משפטים הדומים ביותר מבחינת הייצוג לשני המודלים וגילו גם כאן דמיון רב. נראה מבטיח אך נדרשות בדיקות מקיפות יותר על מודלים נוספים יותר.

## Review 141, Short: DOLA: DECODING BY CONTRASTING LAYERS IMPROVES FACTUALITY IN LARGE LANGUAGE MODELS, 08.09.2023

<https://huggingface.co/papers/2309.03883>



אנחנו משתמשים במודלי שפה למשימות רבות אבל האם לסמן על פלטיהם? עם כל הצער עדין לא. יש תופעה הנקראת הדזיות (hallucinations) של מודלי שפה כאשר מודלים שפה מדברים שטויות. זו בעיה מאוד רצינית בטח אם אתם רוצים לשים מודל שפה בפרויקטן.

از היום ב-#shorthereviewspaperreviews אנחנו סוקרים מאמר המציג פתרון לבעה החמורה זו. המחברים מבססים את שיטתם על התוכנה המעניינת של מודלי שפה שנייה לצפotta כאשר מօסיפים שכבת סופטמקס המחשבת את התפלגות הטוקנים אחרי כל בлок הטרנספורמר.

למעשה יש כאן שתי תופעות נפרדות. עבור טוקנים קלים (יחסית) לניחוש (הנובעים מכך דקוק למשל) אז התפלגות טוקנים משכבות האמצע בערך לא משתנה ושווה להתפלגות הסופית של הטוקנים. בטוקנים היוצרים קשיים הרה��lugות משמעותית כמעט עד השכבה האחרונה – כלומר בשכבה לפני الأخيرة ההתפלגות הטוקנים עשויה להיות שונה מאוד מהשכבה הסופית.

אחד ההסבירים לכך הוא "שואב את 'הידע העובדתי' מהזיכרון שלו" (וליהה על נפנפי הידעים אבל כנה כתוב במאמר). המחברים מציעים לנרגמל את הסופטמקס הסופי עם השכבה בעלת שני הגובה ביותר ( מבחינת התפלגות הטוקנים). ככלומר הסתברות התокן פרופורציונלית להשתנות המקסימלית של ההסתברות של הטוקן זהה (= "כמה הלמידה"?).

עושים זאת רק לטוקנים בעלי הסתברות גבוהה מספיק בשכבת סופטמקס الأخيرة – השכבות עם שינוי מקסימלי נבחנות על סט ולידציה. השוני נמדד במונחי Jensen-Shannon Divergence או DJS (גראסה סימטרית של KL) בין התפלגות הטוקנים. מכיוון שנרגמל זה עלול לדפוק את הדקוק מօסיפים "כנס על חזרתיות" (שלא יפלוט את אותו הטקסט יותר מפעם אחת).

**Input:** Who was the first Nigerian to win the Nobel Prize, in which year?  
**Output:** Wole Soyinka was the first Nigerian to win the Nobel Prize, in 1986.

	$w$	$o$	$s_0$	$y$	$im$	$a$	$wes$	$st$	$first$	$Niger$	$ian$	$to$	$win$	$the$	$Nobel$	$Prize$	-	$in$	$-$	$l$	$g$	$B$	$6$	$-$
10	1.9	0.0	0.03	1.76	0.0	0.0	6.45	0.29	0.07	0.6	0.03	0.48	0.13	0.1	0.02	0.11	2.97	1.84	0.12	0.0	0.0	0.0	7.56	0.23
16	4.76	0.04	0.42	10.5	0.05	0.07	3.65	0.21	0.02	0.67	0.0	0.29	0.17	0.02	0.04	0.02	4.77	1.89	6.13	9.76	12.4	15.16	16.86	0.16
26	11.41	3.15	7.15	12.67	5.28	3.5	1.22	0.08	0.02	0.75	0.0	0.18	0.15	0.12	0.05	0.04	3.77	1.19	4.58	10.56	19.31	18.66	19.67	0.13
28	13.23	8.6	10.01	14.28	8.99	8.44	0.8	0.26	0.02	0.44	0.0	2.51	0.08	7.37	0.06	0.04	2.08	0.71	6.68	18.72	23.84	21.68	21.33	0.1
32	14.26	18.81	11.61	15.7	12.34	9.29	0.75	4.87	0.09	0.24	0.0	2.4	0.09	6.57	0.05	0.02	2.03	0.38	8.27	17.82	22.89	22.98	21.46	2.07
36	10.18	15.95	12.99	16.32	13.52	11.07	1.85	9.78	0.03	0.06	0.04	0.39	0.73	6.28	0.02	0.01	11.43	4.36	9.19	16.84	19.37	20.38	19.45	10.26
38	7.75	15.97	12.59	16.46	14.52	12.25	7.76	8.33	5.15	6.47	2.48	5.73	10.87	7.41	1.29	8.97	13.57	10.99	12.59	14.02	19.57	16.98	15.63	12.9
44	8.99	16.02	12.91	17.45	15.47	15.52	9.0	11.18	10.75	10.97	12.1	11.4	16.52	13.08	10.94	11.86	14.34	12.16	11.7	13.73	19.44	17.05	15.89	13.47
46	9.06	16.18	13.33	17.81	16.74	14.0	10.63	13.03	12.78	12.66	15.07	13.2	16.06	14.71	13.61	13.61	14.09	12.04	14.19	14.4	19.76	17.17	18.28	12.87
52	9.75	16.3	13.47	17.92	16.45	14.94	11.52	13.95	14.11	13.92	15.82	14.23	16.76	15.6	14.81	14.42	14.47	13.48	14.47	15.02	19.44	17.4	16.45	13.57
56	10.22	16.4	13.63	18.3	16.24	15.52	12.4	14.54	14.71	14.2	15.34	14.8	16.78	15.6	15.02	15.06	14.53	13.8	14.13	14.06	19.63	17.7	16.62	13.42
62	10.66	16.51	14.04	18.24	16.2	16.21	12.69	14.42	15.09	14.99	16.02	14.71	16.88	15.57	15.2	15.31	14.44	13.89	14.47	15.15	19.93	17.93	16.81	13.9
68	10.66	16.49	14.2	18.38	16.3	16.62	13.18	14.53	15.4	14.27	17.03	15.44	16.98	15.82	15.43	15.4	14.27	14.16	14.65	15.54	19.79	18.2	17.14	13.92
74	10.65	16.59	14.31	18.53	16.39	16.77	13.43	15.02	15.95	14.57	18.29	15.5	17.29	16.39	16.9	16.34	14.31	14.53	14.69	15.81	19.93	18.36	17.4	14.25
76	10.8	16.69	14.29	18.64	16.74	16.9	13.36	15.23	15.97	14.76	18.68	15.45	17.31	16.71	16.05	16.46	14.58	14.52	14.84	16.02	20.13	18.6	17.67	14.44
80	11.0	16.69	14.51	18.78	16.62	17.09	13.54	15.6	16.47	14.88	19.12	15.88	17.45	16.98	16.26	16.87	14.85	15.34	15.16	16.34	20.46	18.79	17.83	14.95

Figure 2: Jensen-Shannon Divergences between the final 32nd layer and even-numbered early layers. Column names represent predicted next tokens in each decoding step. Row names indicate the layer indices of the early exit layers, from the 0th (word embedding) layer to the 30th layer.

## Review 142: LARGE LANGUAGE MODELS AS OPTIMIZERS, 09.09.2023

<https://huggingface.co/papers/2309.03409>

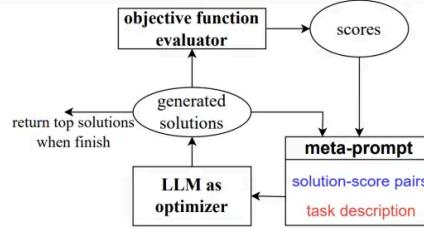


Figure 2: An overview of the OPRO framework. Given the meta-prompt as the input, the LLM generates new solutions to the objective function, then the new solutions and their scores are added into the meta-prompt for the next optimization step. The meta-prompt contains the solution-score pairs obtained throughout the optimization process, as well as a natural language description of the task and (in prompt optimization) a few exemplars from the task. See Figure 3 for a sample meta-prompt for prompt optimization.

מודלי שפה נמצאים היום כמעט בכל מקום: הם עוזרים לנו ל כתוב תוכן, לבצע מגוון משימות הקשורות לשפה טבעיות כמו תרגום, ניתוח סנטימנט, אמן מנהליםアイテム דיאלוגים מעוניינים להנאהנו. אבל האם הם מסוגלים לפתור בעיות אופטימיזציה כמו גרסיה לינארית או בעית איש המכירות המתיל?

היום ב-#shortherebrewpapereviews אנו שוקרים קיצרות מאמר שמודלי שפה כן מסוגלים לפתור בעיות אופטימיזציה הנ"ל, כמובן אם מדובר איטם יפה (כלומר מהנדסים פרומפטים בצורה מתאימה) אז הם מצליחים לפתור בעיות גרסיה לינאריים (במייד אחד, כלומר למצוא שני מקדמים של הישר) וגם בעית איש המכירות המתיל (למצוא מסלול הקצר ביותר המבקר בסט נקודות – כאן על המשיר).

במקרה של גרסיה בוחרים ערכים של שני המקדמים (a,b) של גרסיה דוגמאות 50 נקודות x ומחשבים בהם את ערך הפונקציה בתוספת רעש גaussi. מתחילה כמה זוגות אקרים של a - b שנדרגו. נותנים את השגיאות האלו למודל שפה ומבקשים ממנו לתת ערכים של a - b שemmazzurim את ההפרש הזה.

המודל מנהש ואז מספקים לו כמה זוגות של ערכי  $w$  ו-  $b$  המוצלחים ביותר. והמודל מצליח די מהר להגיע די קרוב לתשובה הנכונה. דבר דומה עושים לגבית איש המכירות המתויל וגם שם מודל שפה די מצליח. אצין לי שלא ברור לי איך מעבירים את הדעתהסט למודל. אחר כך המחברים ביצועי אופטימיזציה של הפרומפט (נקרא meta-prompt) במטרה לסייע את השגיאה על הטסט סט. מטה-פרומפט זה מורכב משני דברים:

1. הפרומפטים הקודמים שנענו והדוקים של הפתרנות (שערוני המקדים) שהמודל סיפק באיטרציות הקודמות.
2. תיאור הבעיה יחד עם הדגימות מהטרין סט ( ממש בחירת מיניבאטץ')

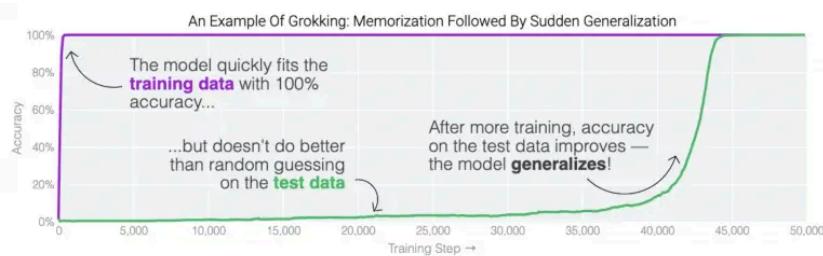
התוצאות די נחמדות...

Table 1: Top instructions with the highest GSM8K zero-shot test accuracies from prompt optimization with different optimizer LLMs. All results use the pre-trained PaLM 2-L as the scorer.

Source	Instruction	Acc
<i>Baselines</i>		
(Kojima et al., 2022)	Let's think step by step.	71.8
(Zhou et al., 2022b)	Let's work this out in a step by step way to be sure we have the right answer. (empty string)	58.8 34.0
<i>Ours</i>		
PaLM 2-L-IT	Take a deep breath and work on this problem step-by-step.	80.2
PaLM 2-L	Break this down.	79.9
gpt-3.5-turbo	A little bit of arithmetic and a logical approach will help us quickly arrive at the solution to this problem.	78.5
gpt-4	Let's combine our numerical command and clear thinking to quickly and accurately decipher the answer.	74.5

## Review 143, Short: Explaining grokking through circuit efficiency, 11.09.2023

<https://arxiv.org/abs/2309.02390>



Source: <https://pair.withgoogle.com/explorables/grokking/>

יש תופעה מעניינת, הנקראת גרוקינג (grokking) המתרחשת (לפעמים) במהלך אימון של רשתות נירונים. למעשה גרוקינג מחלק אימון של רשת נירונים לשני שלבים עיקריים: הראשון הוא שלב ההשינון (memorization) כאשר הרשות משנהנת את הדוגמאות (overfit) ושלב ההכללה (למידה אמיתית).

היום ב-#shorthereviewspaperwehave שוננו סוקרים מאמר המנסה להסביר למה התופעה זו מתרחשת. למה בכלל רשות נירונים "משננת" את הדוגמאות בתחילת האימון מרגע מסוים עוברת למשטר הכלכלי, ככלומר למידה אמיתית.

קודם כל המחברים שמו לב שבסוף משטר השינוי הדיק של המודל הוא כמעט מושלם והלוואן מדויק ולחבות זאת מרגע מסוים הרשות מתחילה ללמידה להקליל ממשום מה למרות זאת. המחברים טוענים שהסבירה לכך היא הרשות מצליחה להוריד את הלוואן עוד יותר (במצב של דיק כמעט מושלם על הטריין סט) על ידי כך שהיא מתחילה להוציא חיזויים בטוחים יותר (מעבר לקטגוריות הנכונות).

אני חשב שראיתי מאמריהם (לדעתי בנושא double descent) שטענים שהירידה בלוטו צו במצב של ערכי ליאו מאד נמנוכים הופכת את הפונקציה שהרשות ממדלת ליוטר פשוטה (מאפס הרבה פרמטרים ומגדיל את האחרים). הם גם מתארים תופעה מעניינת נוספת: כאשר במשטר השינוי מקפאים את לוג'יטים (שמהם מחשבים את ההסתברות הקטגוריות) הנורמה של פרמטרי הרשות עולה עם הגדלת הדאטסהט.

לעומת זאת במשטר הכלכלי נורמה של פרמטרי הרשות לא תליה בגודל הדאטסהט. בנוסף הם שמו לב שלפעמים ניתן לצפות תופעה שהם קראו לה semi-grokking כאשר המעבר במצב שינוי מתרחש אבל הרשות מצליחה להגיע למצב הכלכלי אבל לא מצליחה להגיע להקללה טוביה ונתקעת אייפשו באמצעות (semi-overfit).

בקיצור מאמר מאד מעניין – מומלץ בחום!

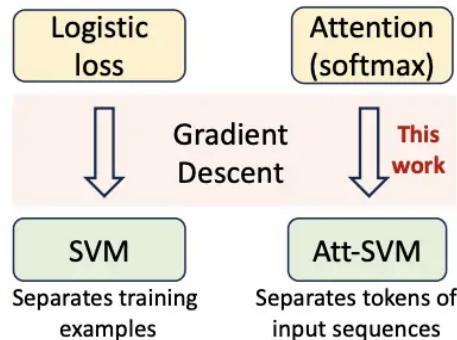


Figure 3: Implicit biases of the attention layer and logistic regression.

## Review 144: Transformers as Support Vector Machines, 12.09.23

<https://arxiv.org/abs/2308.16898.pdf>

המאמר זהה שונה מרוב המאמרים על למידה عمוקה. הוא לא מנסה לשפר איזה מודל או להוכיח ביצועי SOTA במשימה זו או אחרת. הוא מנסה להסביר סודות של ארכיטקטורת הטרנספורמרים שעלייה מבוססים רוב המודלים החזקים של ימינו.

az h'ym b-#shorthereviewspapersthoreviews anno סוקרים מאמר שמנסה לחקור מה למשה מואופטם בטרנספורמרים כאשר אנחנו עושים דרך פונקציית הפסד (נגיד על הטוקן הבא). המחברים מצאו בהקשר המדבר בעית אופטימיזציה של הטרנספורמרים שקופה למודל מסווג Support Vector Machine או SVM בדרך מאד מעניינת. קודם כל מה SVM? זו טכניקה שהומצאה בשנות ה-80 ובמהותה היא מנסה למצוא ישר (או עיקומה מורכבת יותר כאשר מדובר בקורסיל SVM) המרחיק כמה

שיותר בין הנקודות השונות לקטגוריות שונות. אז מה משותף בין אופטימיזציה של הטרנספורמרים לבין SVM שלכאורה נראה נראים דברים לא קשורים כלל. נניח שיש לנו בעית SVM (דו-אלית למי שמכיר אבל זה פרט חשוב). מטרתה למצוא מטריצות  $Q$  ו-  $K$  בעלות נורמה גרעינית (סכום של ערכי עצמים) מינימלית של  $W = QK$  עם ראנק קטן מ- $d$  (מספר השורות ב- $W$ ) שיפורידות את הטוקן האופטימלי משאר הטוקנים (הם חקרו את חיזוי הטוקן הראשון).

ההפרדה גם מתבצעת עם המכפלה של  $K$  ו-  $Q$ . אז מתרברר שהפתרון של בעיה זו (מטריצות  $Q$  ו-  $K$ ) הוא פתרון של בעית מצער של בלוק טרנספורמר אחד (כולל MLP בסוף). כאן בעיה SVM לא קמורה. עוד משפט מעניין שהם הוכיחו שאם עושים רפרמטריזציה של בעית אופטימיזציה של הטרנספורמרים עם  $K = WQ$  ומופיעים את שכבת הטרנספורמר שלנו במנוחה  $W$  זה שקול בפתרון בעית SVM כאשר מזעררים את נורמת פרובניאס של  $W$  (שווה לשורש מ- trace של מטריצה מוכפלת בשחלופה). בקיצור מאמר סופר מעניין ודי כבד מתמטי אבל עדין שווה קריאה.

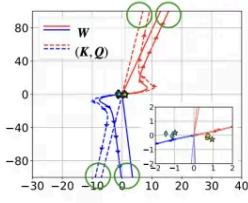


Figure 1: GD convergence during training of cross-attention weight  $W$  or  $(K, Q)$  with data. Teal and yellow markers represent tokens from  $X_1$  and  $X_2$ , while stars mark optimal tokens. Solid and dashed lines depict attention-SVM (Att-SVM and Att-SVM\*, respectively) directions mapped to  $z_1$  (red) and  $z_2$  (blue), with arrows illustrating GD trajectories converging towards these SVM directions. Green circles denote GD  $\leftrightarrow$  SVM pairings.

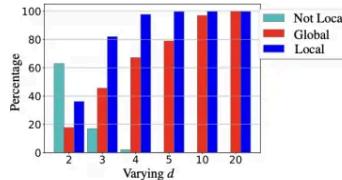


Figure 2: Percentage of different convergence types when training cross-attention weights ( $W$ ) using GD and varying dimension ( $d$ ). Red and blue bars represent the percentages of convergence to globally-optimal and locally-optimal (including global) SVM solutions, respectively. Teal bars are complements of the blue bars. Larger over-parameterization ( $d$ ) increases the likelihood of global convergence.

# Review 145: Textbooks Are All You Need II: phi-1.5 technical report, 13.09.23

<https://arxiv.org/abs/2309.05463.pdf>

## סקירה זו נכתבת על ידי עדן יבין

	Train time (GPU hrs.)	MicroBatch (max)	Inf. speed (per token)	Inf. memory (at 2048 ctx.)	Data size (tokens)	Train tokens
Llama-7B	> 80K	2	14ms	18G	1T	1T
phi-1.5 (1.3B)	1.5K	8	<3ms	3.5G	30B	150B
phi-1.5-web (1.3B)	3K	8	<3ms	3.5G	100B	300B

Table 1: Comparison of compute of different models using a single A100-80G with context length 2048 and fp16.

בזמן האחרון עם שחרור מודלי שפה כמו Falcon-180b, השוגד המודול הוא האינדיקטור העיקרי לביצועי המודל, ככלומר מודל גדול יותר שהוא מודל טוב יותר. חוקרים ממיקורסופט הראו שההבדן נוכן עם שחרור של מודל חדש קטן עם ביצועים מצוינים. איך הם עושים זאת? הימם נגלה ב-#shorthereviewspaper.

מבחן טכנית אין שהוא מיוחד בארכיטקטורה של המודל: טרנספורמר עם 24 שכבות ו-32 ראשי attention. הם השתמשו ב-flash attention לצורך אימון מהיר יותר (לא שהוא מיוחד) והוטקניזר של codegen-mono. אך מה כן מיוחד? ובכן העבודה העיקרית שלהם הייתה המידע לאימון. מבחן טכנית אין שהוא מיוחד בארכיטקטורה של המודל: טרנספורמר עם 24 שכבות ו-32 ראשי attention. הם השתמשו ב-flash attention לצורך אימון מהיר יותר (לא שהוא מיוחד) והוטקניזר של codegen-mono.

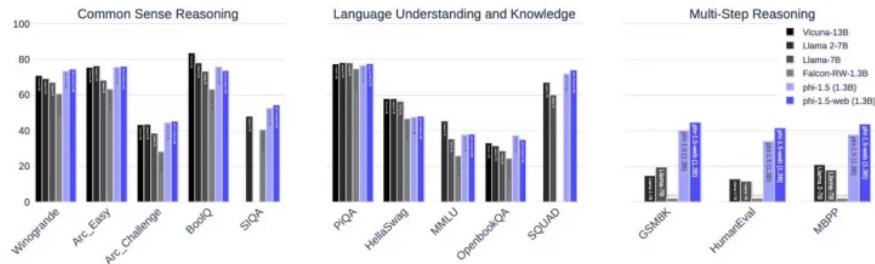
אך מה כן מיוחד? ובכן העבודה העיקרית שלהם הייתה המידע לאימון. המידע לאימון הורכב מ-7 מיליארד שהי שייכים לדאטה סט של המודל הקודם שלהם – phi-1. אלו הם הוסיףו עוד 20 מיליארד טוקונים בצורת ספר לימוד(textbook) על 20 אלף תחומים שונים כדוגמת תאריות המות, פעילותיות יומיומית ועוד. לא הוצג כיצד ספרים אלו נוצרו אלא רק שהם לא נוצרו על ידי אדם.

המטרה בספרים אלו הייתה לשפר את יכולת הסקנת המסקנות והידע הכלול של המודל לגבי העולם. החוקרים ציינו שהם הוסיףו גם 6 מיליארד טוקונים של קוד שבו היחידים בכל הדאטא סט (30 מיליארד טוקונים עד כה) שנכתבו על ידי אדם ולא על ידי מכונה. עם דאטא סט זה הם אימנו את phi-1.5 אך לא עצרו בכך. החוקרים רצו לבדוק את השפעת הוספת מידע שנאסף מהאינטרנט (כלומר נכתב על ידי אדם) ומחקו מידע לא חינוכי על ידי השיטה הבאה:

- שימוש במודל שפה כדי להחליט עד כמה דגימה היא "בעלת ערך חינוכי"
- אימון מסוווג שבהתוצאות נבדיג של דגימה מסוווג את הערך החינוכי
- מחיקה של כל דגימות ללא ערך זה.

חשוב לציין שהחוקרים לא השתמשו בשום שיטה של אימון נוסף (fine-tune) על מעקב אחר הוראות או שיטות RLHF, ובכל זאת הגיעו לביצועים מצוינים מול מודלים שכן עברו את אחד או כל התהילכים אלו. לטענתם, זה רק מוכיח עד כמה הדאטא הוא הדבר החשוב ביותר. סה"כ מדובר טכני ופחות תיאורטי עם פינות קטנות

شمיקרוסופט לא חושפת אף עדין אפשרות ללמוד ממנה לא מעט על הפרטים הטכניים הקשורים לאימון מודלי שפה גדול אלו.



## Review 146: Reinforced Self-Training (ReST) for Language Modeling, 14.09.23

<https://arxiv.org/abs/2308.08998.pdf>

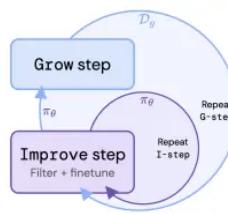


Figure 1 | **ReST** method. During **Grow** step, a policy generates a dataset. At **Improve** step, the filtered dataset is used to fine-tune the policy. Both steps are repeated, **Improve** step is repeated more frequently to amortise the dataset creation cost.

שמעתם על RLHF? ראשית תיבות אלו הפכו להיות מאוד פופולריים לאור הצלחתו של ChatGPT שאומן בשיטה זו. מזמן לא מעט מודלי שפה מאומנים עם השיטה זו למרות שיש טענים שניים שהשיג את אותו הדבר עם (SFT) (self-supervised fine-tuning).

אבל היום ב-#shorthereviewspapereshortthebrewpaperereviews נסקרו מאמר שמציע שיטה לשיפור של RLHF. שיפור הביצועים המיחיל מושג על ידי שיפור אינטואיטת הדעתה-the-fly. אבל קודם כל בואו נבון מה זה בעצם RLHF לאימון מודל שפה יסודי (foundational). הוא תהליך המורכב מ- 4 שלבים:

- הוא אימון מקדים (pretraining) על קורפוס ענק
- יכול מפוקח (SFT) או instruction tuning – אימון נוסף על דאטاهסט מאד אינטואיטי (בד"כ שאלות ותשובות)
- אימון מודל תגמול (reward) משערך את אינטואיטת הפלט(ציוון סקלרי) של מודל שפה

- הרצה של אלגוריתם RL (למשל PPO שהוא אלגוריתם policy-ho אונליין) כדי לגרום למודל שפה להיות "מיושר יותר" עם היציפות שלנו על ידי מKeySpec פונקציית התגמול ותוך שמירת המודל קרוב לזה בסוף שלב 2

כאמור המאמר הנזכר מציע שיטה לשיפור של RLHF הקודם השלב האחרון מבוצע באמצעות אלגוריתם RL **אופליין** שמאפשר שימוש חוזר בדאטה. האלגוריתם מכיל את השלבים הבאים (הם ניטו זאת על משימת התרגום):

1. גנרט דאטה מודול שפה מאומן מהשלב הקודם וחישוב ציון (תגמול) עבור כל זוג של (טקסט, טקסט מתרגם)
2. בוחרים סף של הציון ומסננים את כל הנקודות שהציוון שלהם קטן ממהספ
3. מכילים מודול על הדעתהזה תור שימוש באלגוריתם של אופליין RL (דאטה חדש לא מיוצר) – שזה סוג של למידת חיקוי (imitation learning)
4. מעלים את הסף ומבצעים את שלב 3 עד שהბיצועים על סט אבלוציה נתון מפסיקים להשתפר
5. אם ביצענו שלב 1 לפחות פחות מ G איטרציות חוזרים לשלב 1 אחרת מסייםים

---

Reinforced Self-Training (ReST) for Language Modeling

---

**Algorithm 1: ReST algorithm.** ReST is a growing-batch RL algorithm. Given an initial policy of reasonable quality (for example, pre-trained using BC) iteratively applies **Grow** and **Improve** steps to update the policy. Here  $F$  is a filtering function, and  $\mathcal{L}$  is an loss function.

---

**Input:**  $\mathcal{D}$ : Dataset,  $\mathcal{D}_{eval}$ : Evaluation dataset,  $\mathcal{L}(x, y; \theta)$ : loss,  $R(x, y)$ : reward model,  $G$ : number of grow steps,  $I$ : number of improve steps,  $N$ : number of samples per context

Train  $\pi_\theta$  on  $\mathcal{D}$  using loss  $\mathcal{L}$ .

```

for  $g = 1$  to  $G$  do
    // Grow
    Generate dataset  $\mathcal{D}_g$  by sampling:  $\mathcal{D}_g = \{ (x^i, y^i) \}_{i=1}^{N_g} \text{ s.t. } x^i \sim \mathcal{D}, y^i \sim \pi_\theta(y|x^i) \} \cup \mathcal{D}$ .
    Annotate  $\mathcal{D}_g$  with the reward model  $R(x, y)$ .
    for  $i = 1$  to  $I$  do
        // Improve
        Choose threshold s.t.  $\tau_1 > V_{\pi_\theta}$  for  $V_{\pi_\theta} = \mathbb{E}_{\mathcal{D}_g}[R(x, y)]$  and  $\tau_{i+1} > \tau_i$ .
        while reward improves on  $\mathcal{D}_{eval}$  do
            | Optimise  $\theta$  on objective:  $J(\theta) = \mathbb{E}_{(x,y) \sim \mathcal{D}_g} [F(x, y; \tau_i) \mathcal{L}(x, y; \theta)]$ 
            | Optimise  $\theta$  on objective:  $J(\theta) = \mathbb{E}_{(x,y) \sim \mathcal{D}_g} [F(x, y; \tau_i) \mathcal{L}(x, y; \theta)]$ 
        end
    end
end
Output: Policy  $\pi_\theta$ 
```

---

## Review 147: Generative Image Dynamics

<https://huggingface.co/papers/2309.07906>

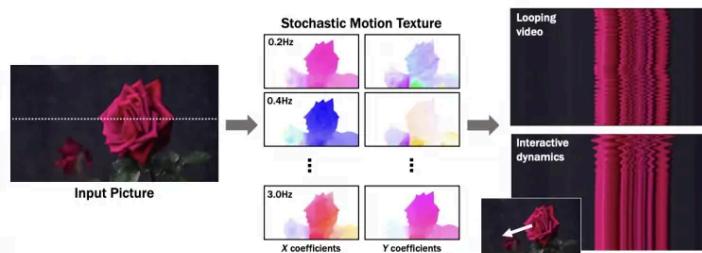


Figure 1. Our approach models a generative image-space prior on scene dynamics: from a single RGB image, our model generates a neural stochastic motion texture, a motion representation that models dense long-term motion trajectories in the Fourier domain. We show that our motion priors enable applications such as turning a single picture into a seamlessly looping video, or simulating object dynamics in response to an interactive user excitation (e.g., dragging and releasing a point on the object). On the right, we visualize the output videos using space-time  $X-t$  slices through 10 seconds of video (along the scanline shown in the input picture).

נתחיל מזה שזה מאמר מאד יפה רק בגלל זה מופיע בו התמרת פוריה (עבדתי בתחום עיבוד אותות כמה שנים יש לי זכרונות חמימים ממנהו). שנית, המאמר עוסק במודל דיפוזיה וביצירת וידאו עם תמונה שאני מאוד אוהב אוק".

از היום ב-shorthereviewspaperreviews סוקרים מאמר ש"מוסיף" תנועה לאובייקט בתמונה, ככלומר מגנרט וידאו קצר המראה אובייקט בדינמיקה (למשל פרח ברוח). יש כמה גישות למידול של תנועה בידאו למשל שדה תנועה שמסמן תזוזה של כל פיקסל בתמונה. ככלומר וידאו המורכב מ-T פרימיטים ניתן לייצוג על ידי  $W \times H \times T$  מספרים לתמונה בגודל  $W \times H$ .

כלומר יש לנו  $W \times H$  מערכים באורך  $T$  מספרים המתארים את המסלול של כל פיקסל בתמונה. אז הנה באה הקטע של התמרת פוריה שאני כה אוהב. במקום לחזות את המערך הזה באו נעשה זאת עם התמרת הפוריה שלו. כמובן שבמקרה שלנו ההתמרה זו מיידית כי כל פיקסל יכול לנوع בשני היכונים (למעשה 4 מספרים ממשים פר פיקסל כי התמרת פוריה היא מרכיבת).

אוק", אז מאמנים מודל דיפוזיה שמטרתו לגנרט התמרת פוריה של תנועתו של כל פיקסל. הבעה אבל שકצת קשה לאמן רשות שחוצה וקטור ארוך (של תדרים) לכל פיקסל בתמונה (לטענת המאמר). אז הם שמים לב שתנועה של כל פיקסל מתוארת בעיקר על ידי תדרים נומכים (תנועה איטית) כאשר התדרים הגבוהים מקבלים ערכים נומכים. אז המחברים מחליטים לקחת רק 16 התדרים הנמכרים ביותר לגנרט (יש איזה נרמול של התדרים כדי למנוע ערכים גבוהים מדי של תדרים מסוימים).

טוב, איך מאמנים מודל דיפוזיה. לוקחים תמונה בדינמיקה (סרטון קצר), מחשבים את שדה התמונה, מעבירים את זה דרך התמרת פוריה זהה מהוות הקטל למודל דיפוזיה שאוטו מריםיהם ואז מסירים את הרעש לאט לאט עם המודל. בנוסף מכניסים למודל את התמונה וגם תיאור טקסטואלי (אופציונלי). ב-*inference* משתמשים את התמרת פוריה של שדה התנועה דרך התמרת פוריה הפעוכה ומרנדרים את הווידאו. והדבר האחרון כמובן משתמשים במודל דיפוזיה לטנטי....

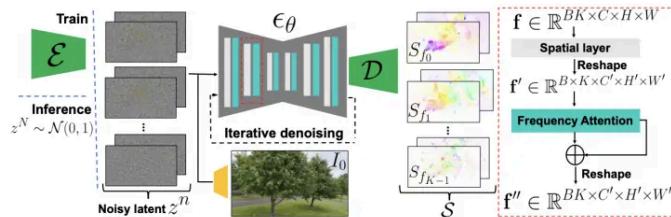


Figure 3. **Motion prediction module.** We predict a neural stochastic motion texture  $S$  through a frequency-coordinated denoising model. Each block of the diffusion network  $\epsilon_\theta$  interleaves 2D spatial layers with frequency cross-attention layers (red box, right), and iteratively denoises latent features  $z^n$ . The denoised features are fed to a VAE decoder  $\mathcal{D}$  to produce  $S$ . During training, we concatenate the downsampled input  $I_0$  with noisy latent features encoded from a real motion texture via a VAE encoder  $\mathcal{E}$ , and replace the noisy features with Gaussian noise  $z^N$  during inference (left).

## Review 148: Knowledge Graph Prompting for Multi-Document Question Answering, 17.09.23

<https://arxiv.org/abs/2308.11730>

סקירה זו נכתבת על ידי עדן יבן

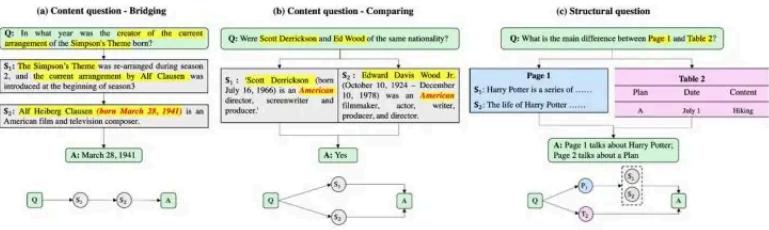


Figure 2: Three popular questions that require reasoning and retrieving over passages/pages/tables from multiple documents. (a) **Bridging questions** rely on sequential reasoning while (b) **Comparing questions** rely on parallel reasoning over different passages. (c) **Structural questions** rely on fetching contents in the corresponding document structures.

אנחנו יודעים שמודלי שפה יודעים לענות על שאלות על גבי מסמכים אם בצורת zero shot או בצורת few shot אבל מה עם המקרה שבו צריך להסתמך על מסמכים שונים כדי לקבל תשובה? או להשוות בין שני מסמכים בעלי מבנה שונה (טבלה וטקסט חופשי)? ספואיר, לא טוב. הימם ב-#showsfewreviews shortherebrewpaperreviews קבלו את הסקירה הבאה:

החוקרים מראים ששיטות קיימות כגון chain of thought שיכולה להתמודד עם הבעיה מביאות אתן מחיר לא קטן. הם אומרים (ולדעתי נכון) ששיטות אלו לא יעילות כי הן צורכות המונח קריאות יקרות למודל השפה. כתוצאה מכל אלו, החוקרים פיתחו שיטה חדשה לה קראו KGP המשמשת בגרף ידע. הם בונים את גרף הידע(Knowledge Graph) שבו קודקודים מסמלים פסקאות / מבנה מסמך וקשר מסוים סמנטי או שייכות מבנה (למשל מסמך A הוא \*סוג\* טבלה). מכאן כאשר מגיעה שאלה חדשה נחלק אותה למצרים. כל מקרה צורך לוגיקה קצת שונה.

1. שאלת על מבנה העמוד, למשל "על מה מדובר בעמוד 1?" כדי לענות על מקרה זה, השיטה משתמשת במודל שפה כדי לסייע את סוג העמוד. למשל במקרה זה מדובר במסמך טקסט, כולם אוסף של פסקאות. לאחר שנדע את סוג המסמך יוכל לבצע הליכה בגרף מקודקוד העמוד אל כל הקודקודים התואמים את סוגו (למשל כל הפסקות במסמך הטקסט).

החוקרים מצינים שהם התייחסו לשני סוגי טקסטים שיכולים להופיע בעמוד והתייחסו לכל אחד באופן שונה: – מסמך טקסט: אוסף של פסקאות, כל פסקה היא קודקוד. – טבלה: התוכן הוא הטבלה עצמה מיוצגת כ-markdown markdown בקודקוד. צין שניתן היה להרחב את השיטה לעוד סוגי של טקסט.

2. מבנה על תוכן עמוד מסוים, למשל: באיזו שנה נולד יוצר הסדרה הסימפסונים? כדי לענות על שאלת זו מבוצע התהליך הבא: מהאזורים את הקודקוד עם המידע הרלוונטי ביותר לשאלת. מוסיפים את המידע הקונטקט לשאלת ומקשים מודל השפה ליצור את המידע הבא שנוצרך כדי לענות על השאלה.

נסתכל על השכונה של הקודקוד בו אנו נמצאים כרגע ונחפש את הקודקוד הדומה ביותר למידע שמודל השפה יצר "צעד הבא". כך נמשיך עד שיש בידינו את כל המידע כדי לענות על השאלה. נשים לב שנitin לבחר יותר מקודקוד אחד בשלב החיפוש, ככלומר יוכל לאחר מכן משיין קודקודים בצד אחד. זוכרים שבתחלת דיברנו על בעיה נוספת הינה מספר קריאות רב למודל השפה (chain of thought)?

ובכן כדי להתמודד עם הבעיה הוציאו לבצע אימון נוסף (tune fine) שבו מודל שפה אומן על מעקב אחר הוראות. במקרה הזה, ההוראה הינה לחזות את הראיה הבאה הרלוונטית כדי לענות על השאלה בהינתן

המידע עד כה. אני יודע שמשמעותם אתכם הפל שבחדר, איך נבנה הגרפ' עצמו? ואובכן נחלק זאת למקרים:

- סוג: באמצעות Extract-PDF מחלצים את סוג המידע במסמך ומוסיפים זאת בתור קודקוד.
  - טבלה: קודקוד של טבלה מכיל את סוג ובנוסף את כל הטבלה c-hunk-markdown.
3. במקרה של טקסט: ראשית נחלק אותו לפסקאות. כל פיסקה מקודדת לשיכון באמצעות מודל אմבידיג או באמצעות TF-IDF. לאחר מכן נחבר בקשר מסמכים דומים. 4. קשותות: בנוסף לקשנות בין פסקאות דומות קיימות קשותות נוספת בין קודקוד העמוד לפסקה הראשונה בו / הטעלה שהוא מכיל בקשר של "מכיל".

T5, Llama, מודלי שפה נוספים מסוגים שונים: MDR ודווקא T5 הביא המון פעמים לחיפוש הטוב ביותר בגרף.

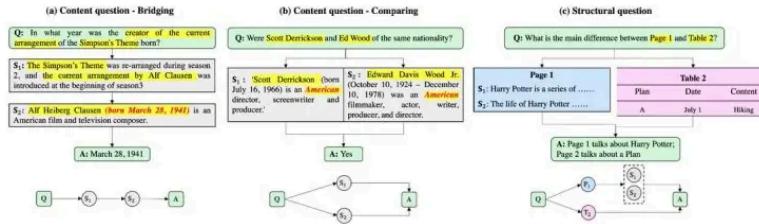


Figure 2: Three popular questions that require reasoning and retrieving over passages/pages/tables from multiple documents. (a) **Bridging questions** rely on sequential reasoning while (b) **Comparing questions** rely on parallel reasoning over different passages. (c) **Structural questions** rely on fetching contents in the corresponding document structures.

## Review 149, Short: Ambiguity-Aware In-Context Learning with Large Language Models, 18.09.23

<https://arxiv.org/abs/2309.07900.pdf>

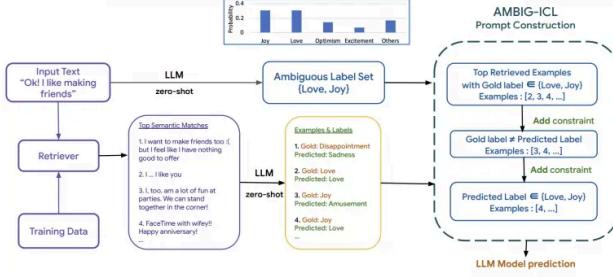


Figure 1: Overview of our proposed method for selecting ICL demonstrations: For each test example, we first use a retriever to rank training data by semantic similarity. At the same time, we identify the ambiguous label set for each test example and also obtain the output predictions on the retrieved training data. Next, we apply three constraints on the top-ranked demonstrations which are: 1) select those demonstrations whose gold label is in the ambiguous label set, 2) select those which are also mis-classified by the model, and 3) select those mis-classified examples whose predicted label is in the ambiguous label set. Finally, we construct prompts with selected ICL demonstrations to get the final model predictions.

למידת **in-context**: מתרבר שמודלי שפה גדולים מסוגלים ללמידה מהקשר (לפעמים אפילו לא צריך הקשר שננקרא **zero-shot**). למשל עבור משימות ניתוח סנטימנטן אTEXM他们是 同时 提供 一个 情感 分类 的 样本 集 和 一个 语义 相关 的 语料 库。 המודל מסתדר בעצמו.

از היום ב-#[shorthereviewspaperreviews](#) סוקרים מאמר המציג שכלול של למידת **in-context** ומציע שיטה לבחירת דוגמת הקשר הטובה ביותר ביחס לדוגמת טסט נתונה. למה זה חשוב? יש דוגמאות די קשות ויש צורך לתת דוגמת הקשר ממש "דומה" כדי לכוון את המודול לכיוון הנכון. למשל עבור קטגוריות קרובות (כמו **joy** ו-**amusement**) או (אכזבה ועצב) מודול שפה עלול להתבלבל ואז חשוב לתת לו דוגמא ש"מסבירה לו" את המשימה בדרך המיטבית ביותר. אז המאמר מציג גישה אלגנטית ואינטואטיבית לבחירה של דוגמת הקשר כזו:

1. בוחרים את הדוגמאות הקשורות ביחס מבחינה סמנטית לדוגמת טסט (עם מודל pretrained המחשב יציג הטקסט).
2. מחשבים את שתי הקטגוריות בעלות הסתברויות הגבוהות ביותר (עם הפרומפט שגורם למודל להפיק הסתברות לכל קטgorיה) עבור דוגמת הטסט ובוחרים מה דוגמאות המתואגות מהשלב הקודם כלו עם אותן הקטגוריות.
3. מדוגמאות מהשלב הקודם בוחרים את אלו שהמודול טעה בהם ומשתמשים בהם בתור דוגמת הקשר.

משתמשים בדוגמאות אלו בשילוב בוצע ניתוח סנטימנטן של מודל עבור דוגמאות טסט (עם פרומפט מהונדו הטוב).

	train	dev	test
EDOS	3,398	486	970
SST	8,544	1,101	2,210
GoEmotions	23,485	2,952	2,978

Table 1: Number of examples in each dataset split.

# Review 150: Language Modeling Is Compression, 21.09.2023

<https://huggingface.co/papers/2309.10668>

Chunk Size	Compressor	Raw Compression Rate (%)				Adjusted Compression Rate (%)			
		enwik9	ImageNet	LibriSpeech	Random	enwik9	ImageNet	LibriSpeech	Random
$\infty$	gzip	32.3	70.7	36.4	100.0	32.3	70.7	36.4	100.0
	LZMA2	23.0	57.9	29.9	100.0	23.0	57.9	29.9	100.0
	PNG	42.9	58.5	32.2	100.0	42.9	58.5	32.2	100.0
	FLAC	89.5	61.9	30.9	107.8	89.5	61.9	30.9	107.8
2048	gzip	48.1	68.6	38.5	100.1	48.1	68.6	38.5	100.1
	LZMA2	50.0	62.4	38.2	100.0	50.0	62.4	38.2	100.0
	PNG	80.6	61.7	37.6	103.2	80.6	61.7	37.6	103.2
	FLAC	88.9	60.9	30.3	107.2	88.9	60.9	30.3	107.2
	Transformer 200K	30.9	194.0	146.6	195.5	30.9	194.0	146.6	195.5
	Transformer 800K	21.7	185.1	131.1	200.1	21.9	185.3	131.3	200.3
	Transformer 3.2M	17.0	215.8	228.2	224.0	17.7	216.5	228.9	224.7
	Chinchilla 1B	11.3	62.2	24.9	108.8	211.3	262.2	224.9	308.8
	Chinchilla 7B	10.2	54.7	23.6	101.6	1410.2	1454.7	1423.6	1501.6
	Chinchilla 70B	8.3	48.0	21.0	100.8	14008.3	14048.0	14021.0	14100.8

מאמר זה משליך ליבי מיד כשראיית את הכותרת. הרעיון לא מעט קשורים בין מודלי חיזוי לבין מודלי כניסה. מאוד מואוד בגדול מודל חיזוי חזק אמרור להפיק את פיצ'רים המהותיים ביותר של הדטה הנחוצים לחיזוי ואולם הפיצ'רים ניתנים לניצול לדחיסת הדטה.

היום ב-#shorthebrewpapereviews סוקרים מאמר מרתק של דיפמיינד שmarcaה שניתן להשתמש במודלי שפה מאומנים בטור דוחשי הדטה. איך הם בכלל ניגשו לבעה זו? הרעיון לא למורי טריוייאלי, איך ניתן להשתמש במודלי שפה לדחיסת הדטה. הרעיון שמודלי שפה (డקורר) יודעים לעשותו הוא לחזות את התפלגות הסטבריות הטוקנים בהינתן הטוקנים הקודמים בסדרה.

از התברר שקיימת שיטה דחיסה שמנצלת הסטבריות אלו לדחיסת הדטה. השיטה נקראת קידוד אריתמטי(arithmetic encoding). עבור פיסת הדטה נתון השיטה מתחילה מאינטראול [0,1]. בהגעה של הטוקן הבא מחלקים את האינטראול לפי ההסתברויות בהתפלגות הטוקן הבא(ברינתן הקודמים) ולקחים את האינטראול המתאים לטוקן הבא בדטה. לאחר הגעת הטוקן האחרון לוקחים את האינטראול שהתקבל ולוקחים ממנו מספר שניית ליצגו במספר המינימלי של ביטים.

סדרת ביטים זו תהווה את הדחיסה של פיסת הדטה שלנו. אז המחברים לקחו את השיטה ובדקו עד כמה ניתן לדחוס דאטסטים שונים מ-3 דומיניום (טקסט, תמונה, אודיו) ודחוסו אותם עם הגישה זו. לאחר מכן הם השוו את התוצאה עם הדוחשים הקלאסיים כמו LZMA2, gzip, -PNG (יש פרטיהם מעוניינים איך המחברים התמודדו עם אורכי הקשר שונים בין מודלי שפה לדוחשים הקלאסיים) ויצא שמודלי שפה מצילחים להביא קצב דחיסה גבוה יותר ב-3 הדומיניום האלו.

אבל תזכיר שיש כאן קאץ' קטן. דוחסים אלו לא לוקחים בחשבון את גודל המודל שנוצר לשומר אותו אם נרצה לפעنه את הדטה. הדוחשים הקלאסיים הם מאוד קטנים ושם זה פחות רלוונטי. זה למעשה מביא אותנו לקונספט מתמטי מורכב הנקרא סיבוכיות קולמוגורוב שמודד את "מורכבות" הדטה בתור סכום של תוכנה (המודל) לדחיסת הדטה וגודלה הדטה אחרי הדחיסה. הם מדברים על זה קצת במאמר – ממליץ להעיף מבט.

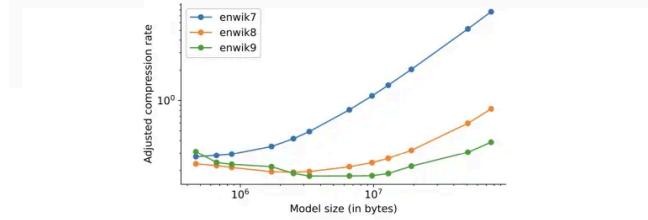


Figure 2 | Adjusted compression rates (compressed size / raw size) for Transformers of different sizes, trained on enwik8 and evaluated on enwik (both axes are logarithmic). Here, the compressed size does not only consider the size of the compressed output (roughly equal to the log-loss) but also the model size, which causes all curves to increase at some point. Every dataset gives rise to an optimal model size, with a good trade-off between performance (the size of the compressed data) and cost of the model (the number of parameters). The larger the dataset, the more parameters we can afford.