

איה פודי

רימא חנא

עיבוד שפות עבודה מס' 4

חלק א:

הרצנו את הקוד הבא בקובץ נפרד, בשביל לשמור את המידע פעם אחת, ובמקום מסוים בשביל להשתמש בנתונים אחר כך

```
load_move_dataset.py x gpt_generation_finetuning.py bert_classifi
1 from datasets import load_dataset
2 dataset = load_dataset("imdb")
3 subset = dataset["train"].shuffle(seed=42).select(range(500))
4 subset.save_to_disk("imdb_subset")
```

מבוקש לקחנו מאגר נתונים בשם "imdb" וקחנו ממנו 500 דוגמאות מ-הדאטא של האימון באופן רנדומלי, ושמרנו אותם בקובץ בשם "imdb_subset".

חלק ב:

עשינו מה מבוקש, בחירתנו עבור ארגומנטי האימון הייתה כך:

```
training_args = TrainingArguments(
    output_dir="./results",
    eval_strategy="no",
    learning_rate=2e-5,
    per_device_train_batch_size=5,
    per_device_eval_batch_size=5,
    num_train_epochs=3,
    weight_decay=0.01
)
```

Num_train_epochs=3:

מספר הדגימות בסט האימון הוא 400, ולכן מעבר אחד בלבד על הנתונים (אפוק אחד) אינו מספיק כדי ללמוד דפוסים מורכבים, כמו אלו הנדרשים למשימת הסיווג שאנו מבצעים. מצד שני, בחירה במספר גדול מדי של אפוקים עלולה לגרום לבעיה של **התאמת יתר (overfitting)**, שבה המודל מתאים את עצמו יותר מדי לנתוני האימון ומאבד את יכולת ההכללה שלו. לכן, בחרנו בערך ביניים, 3, שמאפשר למודל ללמוד דפוסים משמעותיים תוך שמירה על יכולת הכללה על נתונים חדשים.

eval_strategy="no":

כי לא רוצים לשמור את המודל.

Learning_rate=2e-5:

בחרנו קצב למידה זה כי הדאטא שלנו יחסית קטן ומורכב, ומכיוון שמדובר בתהליך Fine-Tuning של מודל מאומן מראש (BERT), קצב למידה קטן כמו $2e-5$ מאפשר התאמה עדינה של המשקלים לדאטא מבלי לגרום לשינויים חדים מדי או לאובדן הידע הקודם של המודל.

weight_decay=0.01:

בחרנו ב $weight_decay=0.01$ כדי למנוע overfitting במהלך האימון. ערך זה מוסיף רגולריזציה שמפחיתה את השפעתם של משקלים גדולים מדי במודל, מה שתורם לשיפור ההכללה של המודל על דאטאות חדשות. הערך 0.01 מתאים כיוון שהוא מספק איזון טוב בין מניעת overfitting לבין שמירה על התאמה לדאטא שלנו.

per_device_train_batch_size=5=per_device_eval_batch_size:

הערך 5 נבחר מתוך איזון בין יעילות חישובית לבין מגבלות הזיכרון של המערכת, תוך שמירה על איכות האימון וההערכה.

עבור קבוצת הבדיקה קיבלנו תוצאת accuracy הבאה:

```
{'eval_loss': 0.3598928912162781, 'eval_accuracy': 0.86, 'eval_runtime': 67.2624, 'eval_samples_per_second': 1.487, 'eval_steps_per_second': 0.297, 'epoch': 1.0}
{'eval_loss': 0.28761555728912354, 'eval_accuracy': 0.87, 'eval_runtime': 74.565, 'eval_samples_per_second': 1.341, 'eval_steps_per_second': 0.268, 'epoch': 2.0}
{'eval_loss': 0.24788598716259893, 'eval_accuracy': 0.9, 'eval_runtime': 73.8156, 'eval_samples_per_second': 1.355, 'eval_steps_per_second': 0.271, 'epoch': 3.0}
{'train_runtime': 3599.151, 'train_samples_per_second': 0.333, 'train_steps_per_second': 0.067, 'train_loss': 0.33868777720133463, 'epoch': 3.0}
240/240 [59:59<00:00, 15.00s/it]
| 20/20 [01:02<00:00, 3.14s/it]
Test Accuracy: 0.90
```

שאלות:

1-הגענו לדרגת דיוק של 0.9, שהיא טובה יחסית אך לא מושלמת, ונציג את הסיבות לכך בהמשך. סוג הסיווג בדאטה שלנו מתבסס על הטון בטקסט: אם הטון חיובי, הביקורת מסווגת כחיובית, ואחרת – שלילית. המודל BERT מצטיין בלמידת הקשרים וסמנטיקה מורכבים, כיוון שהוא אומן מראש על כמויות עצומות של טקסטים מגוונים. באמצעות Fine-Tuning על הדאטה שלנו ובחירת פרמטרים מתאימים, הצלחנו להתאים את המודל לדאטה חדשה מבלי לגרום ל-overfitting-

עם זאת, סיבה אפשרית לכך שלא הגענו לדיוק גבוה יותר היא כמות הדאטה המצומצמת שבה השתמשנו. כמות גדולה יותר עשויה לאפשר למודל להבין הקשרים וסמנטיקה מורכבים יותר. מצד שני, אם היינו לומדים על אותה דאטה יותר מ-3 פעמים, הייתה סכנה ל-overfitting, ולכן איזון זה היה קריטי. דרגת דיוק זו היא התוצאה המיטבית שהצלחנו להשיג תוך שמירה על הכללה ואיזון בין אימון לבין התאמה יתרה.

2- אם נשתמש במודל שאימנו על ביקורות סרטים כדי לסווג ביקורות ספרים, סביר להניח שלא נגיע לדיוק גבוה יותר, אך ייתכן שנשיג דיוק סביר. הסיבה לכך היא שביקורות סרטים וספרים נוטות להיות דומות מבחינת הטון: ביקורת חיובית תתבטא בטון חיובי, ושלילית בטון שלילי. עם זאת, ייתכן שיש הבדלים סמנטיים והקשריים ייחודיים בין ביקורות ספרים לביקורות סרטים.

כיוון שהמודל הותאם לדאטה של ביקורות סרטים ולא נחשף למאפיינים הייחודיים של ביקורות ספרים, יש סבירות גבוהה שהוא יתקשה בזיהוי נכון של ביקורות ספרים. לסיכום, איכות התוצאות תלויה במידת הדמיון בין ביקורות הספרים לביקורות הסרטים שבהן השתמשנו במהלך האימון.

3- אם נשתמש במודל שאימנו על ביקורות סרטים ממאגר אחד כדי לסווג ביקורות סרטים ממאגר אחר, התוצאות יהיו תלויות במידת הדמיון בין המאגרים. לכל מאגר עשויות להיות תכונות ייחודיות, כמו סגנון כתיבה, מבנה משפטיים, או אוצר מילים, המשפיעות על זיהוי הטון.

אם המאגרים דומים במאפיינים המרכזיים שמבדילים בין טון חיובי לשלילי, ייתכן שהמודל יצליח לשמור על דיוק טוב ואף להגיע לרמה גבוהה יותר. לעומת זאת, אם קיימים הבדלים משמעותיים בין המאגרים, ייתכן שהמודל יתקשה לבצע הכללה נכונה, והתוצאות עשויות להיות פחות מדויקות.

לכן, איכות הביצועים תלויה בעיקר בדמיון בין שני המאגרים בתכונות המרכזיות הקשורות לסיווג.

4- כדי לשפר את התוצאות עבור כל אחד מהמקרים, ניתן לבצע Fine-tuning נוסף על המודל שייצרנו, תוך שימוש בדאטה מהמאגרים החדשים אותם אנו רוצים לסווג.

ב-Fine-tuning הראשון, המודל למד להבדיל בין טונים שונים על בסיס הדאטה המקורית. ב-Fine-tuning השני, המודל ילמד את המאפיינים הייחודיים של המאגר החדש (כגון אוצר מילים או סגנון כתיבה ייחודי), וכך יתאים את עצמו למאפיינים אלו. פעולה זו תאפשר למודל להבין טוב יותר את הדאטה החדש ולהגיע לרמת דיוק גבוהה יותר.

חלק ג':

- עבור כל אחד מהמודלים טענו מודל GPT-2 ו-tokenizer
- לקחנו את בדאטה מהקובץ שקיבלנו את שמו.
- חילקנו את הדאטה לפי עמודות ה-label כאשר ה-label=0 שמרנו ב-dataset_label_0 וה-label=1 שמרנו ב-dataset_label_1 ולקחנו מכל אחת 100 דגימות.
- עבור כל אחד משני המאגרים עשינו לו טוקנזציה בעזרת הפונקציה tokenize_reviews עם הטוקניזר המתאים לה.
- עבור כל אחד משני המאגרים חישבנו את אורך הטקסט הממוצע ושמרנו ב-average_length_label_0 ו-average_length_label_1.
- עבור כל אחד משני המאגרים בחרנו ערכי פרמטרי האימון אלה:

```

training_args_label_0 = TrainingArguments(
    output_dir="./results",
    eval_strategy="no",
    learning_rate=2e-5,
    per_device_train_batch_size=5,
    per_device_eval_batch_size=5,
    num_train_epochs=3,
    weight_decay=0.01
)

```

- כאשר כל הפרמטרים שווים לבחירה השלב ב לכן זה אותו הסבר חוץ מ- `eval_strategy="no"`, כי בסיוף האימון אנחנו שומרים את המודל ידנית.
- עבור על מודל יצרנו `trainer` עם הדאטה והטוקניזר והפרמטרים המתאימים לכל אחד.
 - אימנו שני המודלים ושמרנו אותם והטוקנים שלהם בתיקייה שקיבלנו אותה בקלט, המודל הראשון בשם `model_label_0` עם הטוקניזר שלו והשני בשם `model_label_1` עם הטוקניזר שלו.
 - טענו את המודלים והטוקנים שלהם שוב, ועבור כל אחד עשינו יצירת 5 טקסטים. נסביר על כל אחד בנפרד.

עבור המודל של ביקורות שליליות:

```

prompt = "The movie was"
input_ids = tokenizer_label_0.encode(prompt, return_tensors="pt")
attention_mask = input_ids.ne(tokenizer_label_0.pad_token_id)

with torch.no_grad():
    output_label_0 = model_label_0.generate(
        input_ids=input_ids,
        attention_mask=attention_mask,
        max_length=average_length_label_0,
        temperature=1.2,
        top_k=15,
        top_p=0.6,
        repetition_penalty=1.2,
        do_sample=True,
        num_return_sequences=5,
    )

```

ייצרנו 5 טקסטים לפי המודל שאימנו על ביקורות שליליות כאשר מגדירים ההתחלה שלו שהיא "The move was" עם הטוקן המתאים למודל זה, הגדרנו שאורך מקסימאלי יהיה הממוצע של אורכי הטקסטים במודל זה כי יצירת הטקסט היא לפי פרטים שמאפיינים טקסטים אלה ואחד מהם הוא אורך הטקסט, כאשר המידע שנכלל בתוך הטקסט מיוחס לאורכו, לכן היה חשוב שיהיה התאמה.

בגלל שהטקסט שנוצר מתאר דעה מסוימת, צפינו שהוא יכיל מגוון של מילים נרדפות ולא יחזור על אותן מילים שוב ושוב. וגם שלא יחזור על אותן משפטים. לשם כך, הגדרנו `repetition_penalty=1.2` כך שהמודל יקבל "עונש" על שימוש חוזר באותן מילים. מצד שני, רצינו לשמור על עקביות רעיונית באותו הטקסט ולכן לא הגדלנו את הערך יותר מדי. הבחירה הזו איזנה בין גיוון מילולי לשמירה על רעיון עקבי לאורך הטקסט.

מכיוון שרצינו שהמודל ייצור טקסטים מגוונים ושונים בכל פעם, בחרנו ב- `temperature=1.2` ערך זה מעודד את המודל לבחור באפשרויות פחות צפויות מתוך ההפצה שהוא מייצר, וכך מתקבלות תוצאות יצירתיות ומגוונות יותר. עם זאת, שמרנו על ערך מתון ולא גבוה מדי כדי להימנע ממקרים שבהם התוצאה תהיה אקראית ולא עקבית מבחינת ההקשר. הבחירה ב- 1.2 איזנה בין גיוון מילולי ושמירה על הרלוונטיות של הטקסט לנושא המרכזי.

Top_k=15: כאשר המודל בוחר את המילה הבאה, הוא מתמקד ב-15 המילים בעלות ההסתברות הגבוהה ביותר. לאחר ניתוח הדאטה, זיהינו שבכל טקסט יש שני חלקים עיקריים: מידע אינפורמטיבי על הסרט והבעת דעה. המטרה הייתה לשמור על מבנה זה ולהכניס גיוון בבחירת המילים. כאשר הקטנו את `Top_k`, קיבלנו טקסטים שהתמקדו רק במידע האינפורמטיבי, ללא ביטוי של דעה. מצד שני, כאשר מדובר בביקורות שליליות, לרוב הן כוללות טון שלילי ומילים שמדגישות תכונות שליליות. לכן, בחירה ב- `Top_k=15` איזנה בין שני הצרכים – שמירה על מידע אינפורמטיבי ובחירה מגוונת במילים שמייצגות דעה שלילית.

Top_p=0.6: פרמטר זה קובע את סף ההסתברויות המצטבר לבחירת מילים. כפי שהוסבר קודם, טקסטים של ביקורות שליליות לרוב מתארים תכונות שליליות. בחירה בערך 0.6 הייתה מספיקה כדי לתפוס את הטון השלילי של הביקורות, ובו זמנית איזנה בין שמירה על המידע האינפורמטיבי המרכזי לבין גיוון בבחירת המילים. ערך זה הבטיח שהמודל יבחר מילים רלוונטיות ולא יחזור על אותן מילים שוב ושוב, תוך שמירה על קוהרנטיות ומשמעות הטקסט.

עבור המודל של ביקורות חיוביות:

```

input_ids = tokenizer_label_1.encode(prompt, return_tensors="pt")
attention_mask1 = input_ids.ne(tokenizer_label_1.pad_token_id)

with torch.no_grad():
    output_label_1 = model_label_1.generate(
        input_ids,
        attention_mask=attention_mask1,
        max_length=average_length_label_1,
        temperature=1.05,
        top_k=40,
        top_p=0.8,
        repetition_penalty=1.1,
        do_sample=True,
        num_return_sequences=5,
    )

```

בביקורות שליליות הטון הכללי היה שלילי וברובו התרכז בתכונות שליליות, בעוד שבביקורות חיוביות היה גיוון רב יותר, שכלל גם תיאורים חיוביים וגם שליליים עם נטייה לחיוב, למשל ציון הנאה מהסרט. לכן, במודל החיובי בחרנו להגדיל את ערכי **top_p** ו**top_k** כדי לאפשר גיוון רחב יותר בבחירת המילים, שיתאים לאופי המגוון של הביקורות החיוביות, אך מבלי להגדיל יותר מדי ולפגוע ביכולת לשמור על הטון המרכזי והייחודיות של הביקורות החיוביות. כך נשמר איזון בין גיוון לבין קוהרנטיות ומשמעות הטקסט.

בהתאם להגדלת ערכי **top_p** ו **top_k** שהפחיתה את הסיכוי לחזרות בטקסט, הקטנו את ערך **repetition_penalty** ל-1.1 כדי לאפשר חופש יצירתי מסוים במודל מבלי לייצר כפילויות. בנוסף, מכיוון שהגיוון בטקסט גדל, הקטנו את ערך **temperature** ל-1.05 כדי לשמור על עקביות בין הנושאים בטקסט ולהבטיח שהטון והמסרים יהיו ממוקדים וברורים. כך יצרנו איזון בין גיוון לבין שמירה על משמעות הטקסט והטון הרצוי.

שאלות:

1-התוצאות היו לא ברורות ויש לפעמים חוסר עקביות כאשר עוברים מנושה לננושה אחר לא קשור, אבל ניתן לזהות הטון הכללי בטקסט. ניסינו להתאים הפרמטרים ככל שניתן בהתאם לדאטה שלנו אבל קיבלנו תוצאות לא טובות מספיק.

2- במודל של הביקורות השליליות, הטון היה ברור מאוד ומזוהה כשלילי, כשהתיאורים כללו בעיקר תכונות שליליות של הסרט, כמו עלילה משעממת או משחק גרוע. לעומת זאת, במודל של הביקורות החיוביות, חלק מהמשפטים היו לא חד משמעיים, והיה קשה להבחין אם מדובר בטון חיובי או שלילי. עם זאת, היה ברור שנטיית הטקסט היא חיובית, במיוחד כאשר הזכירו ביטויים כמו "הסרט היה טוב", למרות שהתכונות שהוזכרו היו שליליות. זה מצביע על כך שביקורות חיוביות כוללות לא רק תיאורים חיוביים אלא גם תיאורים שליליים, כשבסופו של דבר הדעה הכללית היא חיובית.

3- מהתוצאות שקיבלנו נוכל להסיק שהדאטה לא היה מספיק גדול בשביל לתפוס ההקשרים המסובכים לכל מודל, לכן אם נאמן אותו על הרבה יותר דגימות, נצטרך להקטין מספר האיפוקים כי עם מספיק דאטה אין צורך לחזרה על דגימות, כך המודל יתופס יותר פרטים ועובד יותר טוב מצד אחד, אבל מצד שני זמן האימון יהיה ארוך כי הטקסטים גדולים ומורכבים. ואם נאמן המודל עם פחות דגימות זמן האימון יהיה יותר קטן אבל המודל לא יתפוס ייחודיות בפרטים לכן נצטרך להגדיל מספר האיפוקים כך שילמד על אותו דאטה יותר מפעם אחת, אבל אז יהיה סיכוי יותר גדול ל- overfitting.

4- ה-attention_mask יוצר מסכה שמגדירה אילו חלקים של הקלט חשובים למודל. הערך 1 מציין שהטוקן רלוונטי, בעוד שהערך 0 מציין שהטוקן הוא מילוי (Padding) או לא חשוב. מטרתו להבטיח שהמודל יתמקד רק בטוקנים החשובים ויתעלם מחלקים ריקים או לא רלוונטיים בקלט, כך שהוא יוכל לבצע חישוב ממוקד ויעיל יותר.

5- השימוש ב-prompt מאפשר לדגם להתחיל את הטקסט במשפט פתיחה שנתון, ובכך לתת נקודת פתיחה מובהקת והמשך יצירת תוכן ממוקד סביב נושא מסוים. לכן המשפט הראשון שהמודל שלנו מקבל ליצירת הטקסט הוא "The movie was" מה שמכוון את הדגם להמשיך ולייצר תוכן הקשור לביקורת על סרט, תוך שימוש בנתונים שעליהם הוא אומן. המודל ינסה להשלים את המשפט באופן שמתאים לסגנון ולתוכן של ביקורות שעליהן הוא אומן, וזאת תוך שמירה על הגיון ורלוונטיות לטקסט הנתון שהוא. "The movie was"

עשינו ניסוי ושנינו אותו ל- "My hobby is to" וקיבלנו טקסטים של תיאור תחביבם, כי נקודת ההתחלה הייתה "התחביב שלי הוא" לכן המודל שומר על הקונספט של ההתחלה וממשיך ותאר תחביבים. דוגמה לטיקסת שקיבלנו:

"My hobby is to play a lot of games that have fun and make my mind up. I enjoy watching television shows such as Batman, The Flash and other comedies while listening music with friends or reading books on their shelf when the kids are away for some reason (which makes me love them so much)"

נוכל להסיק שהוא יכל להשלים את המשפט מהמודל המקורי לפני ה-Fine-tuning, שלמד על מגוון טקסטים שונים.

חלק ד:

- עבור כל אחד מסוגי הפורמטים לקחנו המודל והתוקניזר המתאים
- לקחנו את הדאטא ואז חילקנו אותה לפי ערך ה label, ולקחנו 25 דגימות שליליות, ו 25 חיוביות שהן באורך לכל היותר 490 כי בפורמט zero-shot יש מגבלה על אורך הטקסט.
- חיברנו שוב את הדאטא ושמרנו ב- combined_dataset
- עבור כל אחד מהפורמטים ייצרנו את הפריפיקס המתאים וחיברנו אותו בעזרת הפונקצייה add_prefix, ואז עשינו טוקנזציה לחלק של ה text- ואז יצרנו דגימות לפי המודל המתאים, ואז עשינו decode, ואז הפכנו את התשובות מטקסטים למספרים 1\0 בשביל לחשב הדקות
- קיבלנו התוצאות הנ"ל:

Zero-shot Prompting Accuracy: 0.94

Few-shot Prompting Accuracy: 0.92

Instruction-based Prompting Accuracy: 0.92

קיבלנו שהתוצאה הכי טובה עבור zero-shot, ובשניים האחרים קיבלנו את אותה תוצאה, נסיק מזה שהמודל הצליח לסווג יותר כאשר לא נתנו לו יותר מדי הוראות ודוגמאות.

פרומט של zero-shot:

```
zero_shot_prefix = "Request: Tell me if the movie review is positive or negative. The review: "
```

פרומט של few-shot:

```
few_shot_prefix = f"Tag the movie review if it is positive or negative: 1. '{example_label_0}': 'Negative' 2. '{example_label_1}': 'Positive'
```

Now tag this review: " , כאשר example_label_0 דוגמה לביקורת שלילית ו- example_label_1 דוגמה לביקורת חיובית.

פרומט של Instruction-based:

```
Instruction_based_prefix = "You are an expert movie review classifier. Please classify this new movie review as positive or negative: "
```

דוגמה לסיווג מוצלח עבור כל אחת מהשיטות:

Review 1: At least it's not full of senseless violence or fluff. It's also not very full of thought or a smooth " storyline. This story had some potential but the director appears to have lost site of the green. The story is disjointed and goes off in strange

directions, to no conclusion. I also don't believe the director spent much time around a golf country club, either. Some of the depictions are absurd. Not very engaging.

"

```
Review 1 true label: negative
Review 1 zero-shot: negative
Review 1 few-shot: negative
Review 1 instruction-based: negative
```

דוגמה לסיווג לא מוצלח עבור כל אחת מהשיטות:

"

Review 30: Not the most successful television project John Cleese ever did, "Strange Case" has the feel of a first draft that was rushed into production before any revisions could be made. There are some silly ideas throughout and even a few clever ones, but the story as a whole unfortunately doesn't add up to much.

"

```
Review 30 true label: positive
Review 30 zero-shot: negative
Review 30 few-shot: negative
Review 30 instruction-based: negative
```

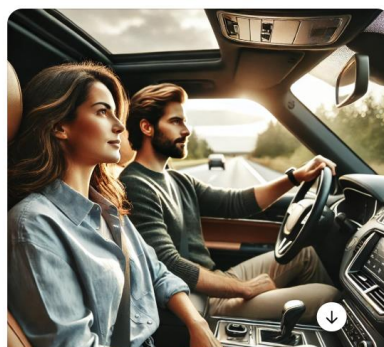
שאלות:

1. התוצאות לא תאמו את הציפיות שלנו, ציפינו שה- **zero-shot** יהי לא דיוק הכי פחות כי לא נתנו לו הוראה מדויקת או דוגמאות מתויגות, אבל בפועל היה לו דיוק הכי גבוה.
2. כן יש סיכוי שהתוצאות ישתנו אם נוסיף יותר מדוגמה, יש אופציה שהדיוק יעלה כי אנחנו נותנים לו יותר מידע ומלמדים אותו יותר לפני שיתחיל לסווג, אבל מצד שני יש אופציה שהוספה זו תסבך את תהליך העיבוד ובשל זה המודל יטעה. לסיכום זה תלוי בדוגמאות שנוסיף וגם באורכי הביקורות.
3. הם דומים בזה ששניהם משתמשים בדאטא קיים וכללי בשביל מסימה ספציפית והשוני ביניהם הוא ש- **Fine-Tuning** : מעדכן את משקולות מודל קיים באמצעות אימון נוסף על דאטה ספציפי למשימה ספציפית. הוא דורש משאבים רבים וזמן חישוב גבוה. ו- **למידה מבוססת פרומפטים** : לא דורשת אימון נוסף, אלא משתמשת בפרומפטים כדי לכוון את המודל. היא גמישה יותר, וחסכונית במשאבים.
4. לא תמיד אפשר להחליף *Fine-Tuning* בלמידה מבוססת פרומפטים, וזה תלוי בסוג המשימה. אם המשימה דורשת שהמודל ילמד מידע חדש שלא היה בדאטא של האימון של המודל המקורי, כמו תחום רפואי או שפה טכנית, *Fine-Tuning* יהיה הכרחי כי הוא מאפשר למודל להתאים את עצמו לנתונים חדשים. מצד שני, אם המודל כבר מכיר את הנושא ואנחנו רק רוצים להנחות אותו איך להשתמש בידע שלו, למשל בסיווג ביקורות או תרגום טקסטים פשוטים, אפשר להשתמש בפרומפטים כדי לכוון אותו בלי הצורך לאמן אותו מחדש. בקיצור, לפעמים פרומפטים מספיקים, אבל במשימות מורכבות שדורשות דיוק גבוה או למידה של מידע חדש-*Fine Tuning* הוא הפתרון הנכון.
5. ניסוח הפרומפט משפיע מאוד על איכות התשובות שהמודל מחזיר, מכיוון שהמודלים מבוססים על דפוסים סטטיסטיים שנלמדו מהנתונים עליהם אומנו. כאשר הפרומפט ברור ומספק הקשר מתאים, המודל מפיק תוצאות מדויקות יותר, ואילו ניסוח מעורפל עלול להוביל לתשובות שגויות או לא רלוונטיות. אפילו שינוי קטן בניסוח יכול לשנות את אופן ההבנה של המודל ולגרום לתשובות שונות. לכן, כדי לשפר את הביצועים, נבצע סיווג תוך שימוש בכמה פרומפטים שונים ונבדוק איזה מהם מניב את רמת הדיוק הגבוהה ביותר.

חלק ה:

1. כן מדובר ב bais סביר להניח שהדאטא שהמודל למד עליה לא הכילה נשים בנהיגה, לכן המודל לא ידע ליצור תמונה כזו.
2. הסיבות שגרמו לו להפיק את תוצאות אלה היא שהמודל עושה מה שלמדו אותו אז במקרה שתהיה בקשה ליצור תמונה שיש בה פרטים שלא ראה קודם הוא לא ידע לייצר משהוא דומה. במקרה שלנו נהוג שכאשר יש משפחה שנוסעת באוטו אז הגבר הוא נוהג, לכן לא נספק מספיק דאטא למודל שמראה אישה נוהגת ולידה גבר לכן קיבלנו תוצאות אלה.
- 3.

תייצר לי תמונה שמראה זוג שנוסעים באוטו כאשר האישה נוהגת ולידה יושב גבר



Here is the image of a couple driving in a car, with the woman driving and the man sitting beside her as the passenger. You can view and download the image if you wish.

4. **שאלת בונוס:**
דוגמה מ- ChatGPT כאשר ביקשתי ממנו שיראה אישה שמציעה נישואין לגבר וזו הייתה התוצאה:

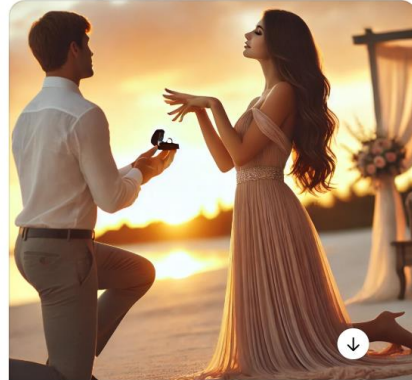
תציג לי תמונה של אישה מציעה נישואין לגבר



Here is the image of a woman proposing marriage to a man in a romantic outdoor setting. You can view and download the image if you wish.

בדרך כלל נהוג שהגבר יציע נישואין לאישה לכן שאימנו את המודל של ChatGPT סופקו רק דוגמאות של גבר שמציע נישואין, לכן הוא לא ידע לייצר תמונה כאשר האישה הוא המבקשת. וכשנשינו לדייק הבקשה קיבלנו תוצאות כזו:

תייצר לי תמונה של אישה כורעת ומחזיקה טבעת כדי להציע נישואין לגבר
והגבר עומד



Here is the image of a woman kneeling and holding a ring to propose marriage to a man who is standing. You can view and download the image if you wish.