תרגיל 1 – שיר עוזיאל 316536580

https://github.com/ShirUziel1/ANLP_EX1.git

שאלה 1

QuAC-CS - הוא סט של שאלות ותשובות, שמטרתן להעריך את היכולת לפתור בעיות של coreference resolution. משימה זו נחשבת למשימה אינטרינזית כיוון שהיא בודקת את היכולת הבסיסית להבין הפניות ואזכורים בתוך הטקסט. המודל נדרש לזהות כינויי גוף, כמו "she" או "he", ולהבין למי הם מתייחסים בהקשר הנתון. יכולת זו חיונית להבנת הקשרים הלוגיים והסמנטיים בין משפטים, מכיוון שהיא מאפשרת לעקוב אחר רצף האירועים ולהבין את המשמעות הכללית של הטקסט.

TORQUE – הוא סט של שאלות ותשובות שמטרתן לבדוק את הבנת רצפי זמן והקשרים בין אירועים בטקסט. המידע נועד להעריך את יכולת המודל להבין את הסדר הכרונולוגי שבו התרחשו האירועים ולזהות את הקשרים הזמניים ביניהם. תכונה זו נחשבת לתכונה אינטרינזית , כי הבנת רצף הזמן היא יכולת לשונית בסיסית וחשובה להבנת הטקסט. כדי להבין את המשמעות הכללית של טקסט, חשוב לדעת מתי כל אירוע התרחש ביחס לאחרים.

QAMR - הוא סט של שאלות ותשובות ,שמטרתו לפרק טקסטים לשאלות המייצגות את המשמעות הסמנטית של רכיבי הטקסט. כל שאלה בסט נתונים זה מתמקדת בהבנת הקשרים בין ישויות ,פעולות ויחסים בתוך הטקסט. המשימה נחשבת אינטרינזית כי היא בודקת את היכולת של המודל להבין את המשמעות הסמנטית וההקשרים הפנימיים של הטקסט, דבר שמסייע ביצירת ייצוגים סמנטיים שיכולים לשמש כבסיס להבנת השפה. המשימה מתמקדת בהבנה של הטקסט, שהיא חיונית להבנה כללית של השפה, קשרים סמנטיים וטקסטואליים.

שאלה 2 a :

1. **Inference-Time Reasoning Techniques**
   a. **Chain-of-Thought**
      i. **תיאור השיטה:** שיטה מאפשרת ביצוע הסקות מורכבות באמצעות שלבי חשיבה לאורך התשובה. ניתן לשלב גם few-shot prompting כדי להשיג תוצאות טובות יותר במשימות מורכבות יותר הדורשות חשיבה מקדימה לפני המענה.
      ii. **יתרונות:** היתרון המרכזי הוא שבמקום לחשוב על כל השאלה ולענות רק תשובה סופית, המודל חושב בצעדים קטנים יותר. השיטה מאפשרת למודל לפרק בעיות מורכבות לצעדים פשוטים, משפרת קוהרנטיות לוגית ותוצאות בבעיות אריתמטיות וסימבוליות, כמו כן ניתן להשתמש ב־ CoT ללא צורך באימון נוסף של המודל, אלא על ידי מתן הדגמות בפרומפט. התוצאות נוטות להיות מדויקות יותר במשימות אריתמטיקה, ניתוח טקסט ומשפטי הגיון. קיימת בשיטה זו גמישות,

היא מתאימה למגוון משימות: פתרון בעיות מתמטיות, שאלות טריוויה, הבנת הנקרא, תכנון פעולות וכו.

iii. **צוואר בקבוק:** אורך הפרומפט והתשובה גדל באופן משמעותי, מה שמעלה את עלות וזמן חישוב הטוקנים.יש צורך ביותר זיכרון. אם הבעיות מורכבות ויש צורך בהסברים רבים יתכן שהעלות תיהיה גדולה.

iv. **מקביליות :** יצירת שרשרת אחת היא פעולה סדרתית, שכן כל שלב תלוי בתוצאה הקודמת לכן לא ניתן למקבל, אך אם רוצים לייצר מספר שרשראות אפשר להריץ אותן במקביל בכמה תהליכים.

b. Tree-of-Thought

i. **תיאור השיטה:** השיטה מייצרת עץ של אפשרויות, כאשר כל "ענף" מייצג שלב ביניים אפשרי .מתחילים בשורש (השאלה המקורית) ומרחיבים מספר צעדים קדימה. בכל רמה מעריכים את "איכות" הצמתים באמצעות מאמת. רק הנתיבים שנתנו תשובה טובה לפי הבדיקה שנעשתה ממשיכים להתרחב למשך עומקי העץ.בסיום, התשובה הסופית נבחרת מתוך העלים עם הציון הגבוה ביותר.

ii. **יתרונות:** מאפשר חיפוש שיטתי רחב, ובכך מגדיל את הסיכוי למצוא את הפתרון האופטימלי.טוב במשימות קומבינטוריות שבהן קיים מגוון רב של מסלולים אפשריים. יש אפשרות לשלב דירוג ביניים כדי לבחור את ההתרחבות.

iii. **צוואר בקבוק:** בכל צומת נדרש קריאה נוספת למודל כדי לייצר ולהעריך אותו. צריך זיכרון לאחסון העץ כולו, כולל ציוני הביניים. ניהול וסינון הצמתים בתוך הזיכרון עשוי להפוך למורכב מאוד.יתכן שהמאמת לא יהיה מספיק טוב או מדויק. עצים עמוקים מאוד יגדילו מאוד את זמן הריצה.

iv. **מקביליות:** אפשר להריץ במקביל את כל הצמתים באותה רמה .

2. Prompt-Based Compositionality

a. Self-Ask Prompting

i. **תיאור :**המודל שואל את עצמו שאלות ביניים, עונה עליהן, ואז מרכיב את התשובות למענה הסופי

ii. **יתרונות:** הפירוק לשאלות תת-משימתיות מאפשר ניתוח שיטתי של בעיות מורכבות, משפר את הדיוק, ומגביר את היכולת להתמודד עם משימות הדורשות הבנה מרובדת והסקה מרובת שלבים.

iii. **צוואר בקבוק:** מגביר משמעותית את מספר הפניות למודל בכל שאלה

iv. **מקביליות:** ניתן להריץ את השאלות הביניים במספר תהליכים מקבילים, במיוחד כאשר הן מתייחסות לדוגמאות שונות, וכך לקצר את זמן העיבוד הכולל.

b. Least-to-Most

i. **תיאור:** מדובר בשיטה דו-שלבית. בשלב הראשון המודל מפרק את הבעיה המורכבת לתת-בעיות פשוטות יותר, ובשלב השני פותר את התת-בעיות בסדר עולה מהפשוטה למורכבת, כשכל פתרון מזין את הבא.

ii. **יתרונות:** השיטה מאפשרת התמודדות עם בעיות מורכבות שהמודל לא יכול לפתור ישירות, על ידי בניית פתרון הדרגתי. הפירוק השיטתי משפר את הדיוק בבעיות רב-שלביות ומאפשר למודל "לבנות" ידע באופן מדורג, כשכל צעד מתבסס על הקודם.

iii. **צוואר בקבוק:** דורש מספר פניות למודל - פעם אחת לפירוק הבעיה ופעמים נוספות לפתרון כל תת-בעיה. כמו כן, דורש יכולת תכנון מראש טובה כדי לזהות את סדר הפתרון הנכון מהפשוט למורכב.

iv. **מקביליות:** מוגבלת - בעוד שניתן להריץ במקביל את שלב הפירוק לתת-בעיות, פתרון התת-בעיות חייב להיות סדרתי כי כל שלב תלוי בתוצאות הקודמים. זה מגביל את היכולת לנצל עיבוד מקבילי.

3. Test-Time Compute Scaling

a. Self-Consistency

i. **תיאור השיטה:** היא גישה לפיה שואלים את המודל את אותו הפרומפט מספר פעמים, ולאחר מכן בוחרים את התשובה שמופיעה הכי הרבה פעמים כתשובה הסופית. כלומר התשובה הסופית מבוססת על הצבעת הרוב . לרוב זה מוביל לתשובה מדויקת יותר בהשוואה לתשובות בודדות .

ii. **יתרונות** השיטה מסייעת לשפר את תשובות המודלים במיוחד בבעיות מתמטיות, לוגיות ומשימות שמצריכות reasoning .
כאשר לוקחים את התשובה שמופיעה הכי הרבה פעמים אנחנו מפחיתים את ההשפעה של תשובות שגויות ולא הגיוניות, ובעזרת פעולה זו אנו יכולים לקבל תשובות סופיות מדויקות יותר.
כמו כן שיטה זו לא דורשת Fine-Tuning ניתן להפעיל אותה על מודל קיים ללא צורך באימון נוסף, כלומר זהו שיפור שנעשה בשלב inference בלבד.
בנוסף קיימת גמישות בשיטה זו מכיוון שהיא מתאימה למגוון משימות: פתרון בעיות מתמטיות, שאלות טריוויה, הבנת הנקרא, תכנון פעולות וכו.

iii. **צוואר בקבוק:** זמן ריצה גדול משמעותית זאת מכיוון שכל שאילתה דורשת עשרות הרצות של המודל. קיים שימוש גבוה בזיכרון מכיוון שאחסון מקבילי של כל התשובות יכול ליצור עומס בזיכרון, במיוחד כאשר כל

תשובה כוללת הסבר ארוך. יתכן מצב של יעילות נמוכה אם אין שונות בין התשובות אם כל התשובות דומות, לא יהיה יתרון להצבעה.העלות יכולה להיות גבוהה.

iv. **מקבול** :
שלב יצירת התשובות- כל תשובה נוצרת באופן בלתי תלוי, ולכן ניתן להריץ את כל המסלולים במקביל על גבי GPU שונים או מעבדים שונים.
שלב בחירת התשובה הסופית- majority vote לא מצריך משאבים רבים.

b. Inference-Time Compute Budget Scaling

i. **תיאור השיטה:**שיטה זו מנצלת תקציב חישוב קבוע כדי להפעיל מספר ריצות ממודל קטן יותר במקום ריצה אחת ממודל ענק. לכל ריצה מייצרים תשובה (או שרשרת מחשבה), ואז בוחרים את התשובה "הטובה ביותר" בעזרת מאמת אוטומטי (למשל בדיקות יחידה או קריטריון רוב)

ii. **יתרונות:** תחת אותו תקציב חישוב, מספר ריצות של מודל קטן 7B–13B יכול להתחרות או לעלות על ריצה אחת של מודל גדול(70B) מאפשרות יותר ניסיונות במחיר נמוך יותר.

iii. **צוואר בקבוק** : צריך מאמת טוב, אם המאמת מורכב או כבד הוא דורש משאבים נוספים, צריך זיכרון לאחסן את כל התשובות, צריך להריץ את המודל הרבה פעמים

iv. **מקבול:** ניתן להריץ כל דגימה והמאמת שלה במקביל על פני GPUs נפרדים.

c. Inference-Time Output Length Scaling

i. **תיאור השיטה:** במהלך השלב של ה־ inference מרחיבים את המגבלה על כמות הטוקנים שהמודל יכול לייצר, כך שהוא מקבל אפשרות רחבה יותר לייצר תשובה .

ii. **יתרונות :** מתן חופש ביטוי גדול יותר לCOS ארוכה יותר, תכנון מראש ותיקון עצמי בזמן אמת.
ניתן לבצע batching של בקשות מרובות ולהריץ מספר פלטים במקביל, ואז לבחור את התוצאה האופטימלית

iii. **צוואר בקבוק:** זמן העיבוד והזיכרון גדלים ביחס ישר למספר הטוקנים. הזמן הנדרש למודל גדל ליניארית ביחס לאורך הפלט, יכול להיות יקר, יש סיכון שהמודל לא ידע מתי לעצור.

iv. **מקבול:** לא ניתן למקבל את יצירת הטוקנים עצמם אך ניתן לשלב אותה עם שיטות אחרות כמו batching להריץ מספר פלט במקביל ואז לבחור את הטובה ביותר.

d. Verifiers –

i. **תיאור השיטה:**מאמתים יודע לקבל תשובה להגיד אם היא נכונה או לא, התחום של לבנות מאמתים גדל. לדוגמא אפשר לבנות כלים אוטומטים, ביטויים רגולרים וכו . שיטה זו מבוססת על הפרדה בין מודל שפותר את השאלה הנשאלת לבין המודל מאמת. במקום להסתפק בתשובה אחת של המודל או לבחור את הנפוצה ביותר במאמת משתמשים במודל/כלי נפרד שמעריך איזו מהן היא האמינה והנכונה ביותר.

ii. **יתרונות:**טכניקה זו טובה במיוחד במשימות הדורשות רמת הסקה גבוהה או בדיקה של עקביות פנימית. נבחרת התשובה בעלת הציון הגבוה ביותר לפי המאמת או שיתבצע שקלול הסתברויות לתשובה הטובה ביותר.

המאמת יודע לזהות ולתגמל תשובות עקביות ונכונות לוגית, גם אם אינן התשובות הנפוצות ביותר. בשונה מ self-consistency, אין הנחה שהתשובה הנכונה היא בהכרח הנפוצה ביותר. המאמת יכול להחזיר לא רק בחירה בודדת, אלא גם דירוג או הסתברות, מה שמאפשר לשים threshold או לקבוע שאין תשובה טובה מספיק. ניתן לאמן Verifier כללי או ייעודי לתחום מסוים – מתמטיקה, ביולוגיה, חוק וכו׳ לדוגמא אפשר לייצר רק את ההתחלות של השרשרת של COS ולבנות סוג של עץ חיפוש ואז להמשיך. להריץ מאמת על ההתחלה שיצרנו ואז להמשיך את העץ . אם יש מאמת טוב זה מאוד מעויל, אבל זה לא קל לעשות אחד טוב במיוחד ככל שהבעיה תיהיה יותר מסובכת.

iii. **צוואר בקבוק**:צורך באימון נוסף של מאמת- לעיתים יש צורך לאמן מודל נוסף שיאמץ את התשובות של המודל הראשון. אימון מודל נוסף דורש זמן ומשאבים נוספים. כמו כן ישנה עלות חישובית גדולה ותלות באיכות המאמת, מאמת לא מדויק או לא מותאם למשימה עלול לשגות בזיהוי תשובות נכונות, ולבחור תשובות לא מדויקות. ולכן צריך שהמאמת שלנו יהיה מדויק, וזאת משימה לא קלה במיוחד שהמשימה הופכת למורכבת. צריך להחזיק בזיכרון גם את המודל הראשי וגם את המאמת וכן יש צורך לשמור את כל הפתרונות המועמדים עד לסיום האימות.

iv. **מקבול:** כל פתרון שנוצר על ידי המודל נוצר באופן בלתי תלוי, ולכן ניתן להריץ אותם במקביל. מריצים את המאמת על כל תשובה בנפרד, מה שמאפשר פריסה של ההערכה במקביל.אולם יתכן שמאמתים שבודקים שלב אחר שלב בפתרון פחות יתאימו להרצה במקביל, או כאשר יש תלות בין הבדיקות השונות.

סעיף b.

בהינתן GPU יחיד עם זיכרון גדול בשביל משימות הסקה מדעיות בפנייה אחת למודל נשלב COS prompting ונבקש פלט מוארך שיאחד בתוכו את שלבי העיבוד הבאים:

1. Planning
2. Backtracking
3. Self-Evaluation

בגלל שיש לנו הרבה זיכרון נוכל לשמור את על התהליך החשבתי של המודל, הזיכרון המרובה מאפשר למודל לזכור את כל השלבים הקודמים שלו, מה שחיוני כשהוא צריך לחזור אחורה ולתקן גישות שלא עבדו, או להתייחס לרעיונות קודמים בשלבים מאוחרים יותר של הפתרון. ראינו בכיתה כי קיימת קורלציה ישירה בין אורך הפלט לאיכות ההסקה של המודלים וברגע החזרה אחורה והערכה העצמית המודל יכול לזהות טעויות בדל שלו. מכיוון שיש לנו זיכרון רב נוכל לייצר פלט ארוך שיכלול את כל שלבי העיבוד וייסע למודל לפתור בעיות הסקה מדעיות.

שאלה 2

- כן, קיבלתי שהמודל שקיבל את התוצאות הכי גבוהות על הווליד‌ציה גם קיבל את התוצאות הכי גבוהות גם על test set

ep2_lr1e-05_bs8: val=0.8015 | test=0.7913

ep3_lr2e-05_bs16: val=0.8407 | test=0.8272

ep4_lr3e-05_bs32: val=0.8333 | test=0.8272

ep5_lr5e-05_bs16: val=0.8750 | test=0.8464

Best validation: ep5_lr5e-05_bs16 (val_acc=0.8750)

Best test : ep5_lr5e-05_bs16 (test_acc=0.8464)

- **אפיון טעויות:**
  - זיהוי ערכים מספריים :
    - המודל היותר חלש התקשה לזהות את המשפטים שהיו עם אותו רעיון אבל הכילו ערכים מספריים

לדוגמא :

It has a margin of error of plus or minus three to four percentage points.	In the 2002 study , the margin of error ranged from 1.8 to 4.4 percentage points.

- המודל היותר חלש התקשה לזהות משפטים עם רעיון שונה שהכילו סימנים מתמטים/אחוזים.

GE 's shares closed at $ 30.65 on Friday on the New York Stock Exchange.	GE stock closed at $ 30.65 a share , down about 42 cents , on the New York Stock Exchange.

  - משפטים עם כמה חלקים
    - המודל היותר חלש התקשה לזהות ולהבין משפטים שמורכבים מכמה חלקים והיה בהם שינוי בסדר החלקים. כמו כן המודל התקשה לזהות אזכורים במשפטים, דבר שהקשה עליו להבין שהמשמעות זהה.

During difficult times for technology venture capital , Vivace raised over $ 118 million in three rounds of venture financing .	Vivace was founded in 1999 and has raised over $ 118 million in three rounds of venture financing .

- המודל היותר חלש התקשה לזהות מונחים טכנולוגים ורפואיים .

The premium edition adds ISA Server , SQL Server and a specialized edition of . BizTalk 2004	The premium edition adds OfficeFront Page 2003 , Acceleration Server 2000 , . and SQL Server 2000