תרגיל 5 – Transformer and LLMs

מבוא

בתרגיל זה נעסוק ברשתות נוירונים חוזרות (RNNs), בטרנספורמרים ובמודלי שפה גדולים (LLMs), דרך משימות שקשורות לsentiment analysis/classification (סיווג רגשות), ג'ינרוט טקסט ועוד.

לשם כך נשתמש בספרייה HuggingFace. תוכלו להוריד ולהתקין אותה באמצעות הפקודה:

```
pip install transformers datasets
```

חלק א': טעינת מאגר הנתונים IMDB movie review

השתמשו במאגר הנתונים IMDb movie review הזמין בספריית המאגרים של Hugging Face. מאגר זה מכיל ביקורות סרטים המסומנות כחיוביות או שליליות.
לשם כך, השתמשו בפקודות הבאות:

```
from datasets import load_dataset
dataset = load_dataset("imdb")
```

בחרו באופן רנדומלי subset מתוך המאגר באופן הבא:

```
subset = dataset["train"].shuffle(seed=42).select(range(500))
```

שמרו את הsubset שלכם לשימוש חוזר:

```
subset.save_to_disk("imdb_subset")
```

על מנת לטעון אותו שוב מהדיסק:

```
from datasets import load_from_disk

subset = load_from_disk("imdb_subset")
```

<u>הערה:</u>
בכל אחד מקבצים להגשה יש לבדוק אם הsubset כבר קיים בנתיב המבוקש (המתקבל בקלט) – אם לא, אז להוריד ולטעון אותו מהספריה, ואם כן אז לטעון אותו מהדיסק, כפי שמתואר לעיל.

חלק ב': Fine-Tuning של BERT לסיווג רגשות

בצעו Fine-Tuning למודל BERT שהורד מראש מ-Hugging Face לצורך סיווג רגשות על הsubset של מאגר הנתונים IMDb. השתמשו ב-API של Hugging Face Trainer.

לשם כך:

1. טענו את המודל *bert-base-uncased*.
2. בצעו טוקניזציה למאגר הנתונים שלכם.
3. חלקו את מאגר הנתונים לסט אימון ואיבלואציה.
4. אמנו את המודל באמצעות Trainer. הסבירו בדו"ח את בחירותכם עבור ארגומנטי האימון.
5. הדפיסו למסך את רמת הדיוק (accuracy) על קבוצת הבדיקה (test set). **וגם דווחו עליה בדו"ח.**

הערות:

1. שימו לב שתצטרכו לשנות את שם העמודה "label" ל"labels".
2. אני ממליצה לשמור את המודל כדי שלא תצטרכו לאמן יותר מפעם אחת **(אך ורק בזמן הפיתוח, אין לשמור מודלים בקובץ שמוגש).**

ענו על השאלות הבאות:

1. האם התוצאות היו טובות? האם הן תאמו לציפיות שלכם? הסבירו.

נניח והאימון הצליח והמודל למד לסווג בצורה טובה דוגמאות מתוך IMDB.

2. אם נרצה לסווג ביקורות **ספרים** באמצעות אותו מודל שאימנו, האם לדעתכם התוצאות יהיו טובות? הסבירו.
3. אם נרצה לסווג ביקורות **סרטים ממאגר אחר**, שאינו IMDB, באמצעות המודל שאימנו, האם לדעתכם התוצאות יהיו טובות? הסבירו.
4. עבור כל אחד מהמקרים מסעיפים 2,3, תארו מה נוכל לעשות כדי לשפר (עוד יותר) את התוצאות?

חלק ג': ג'ינרוט ביקורות סרטים חדשות באמצעות GPT2

בצעו Fine-tuning למודל GPT-2 בנפרד על הביקורות הטובות ועל הביקורות הרעות מתוך מאגר IMDB, כך שיהיו לכם שני מודלים חדשים – אחד עבור כל סנטימנט. כל מודל ילמד לג'נרט ביקורות בהתאם לסנטימנט שאומן עליו. השתמשו במודלים שאימנתם **כדי לג'נרט 5 ביקורות חיוביות ו5 ביקורות שליליות בעזרת המודלים המתאימים**. השתמשו תמיד בprompt ההתחלתי הבא: "The movie was".

לשם כך:

1. טענו את המודל GPT-2 ואת הtokenizer המתאים לו:

```
from transformers import GPT2LMHeadModel, GPT2Tokenizer, Trainer,
TrainingArguments
model = GPT2LMHeadModel.from_pretrained("gpt2")
tokenizer = GPT2Tokenizer.from_pretrained("gpt2")
```

2. השתמשו בפונקציה הבאה או דומה לה, לביצוע טוקניזציה לסט האימון:

```
def tokenize_reviews(dataset, tokenizer, max_length=150):
    if tokenizer.pad_token is None:
        tokenizer.pad_token = tokenizer.eos_token

    def tokenize_function(examples):
        return tokenizer(examples['text'], padding="max_length",
truncation=True, max_length=max_length)

    # Tokenize and rename 'label' to 'labels'
    tokenized_dataset = dataset.map(tokenize_function, batched=True)
    tokenized_dataset = tokenized_dataset.rename_column("label", "labels")  #
    return tokenized_dataset
```

3. בחרו את ערכי פרמטרי האימון training_args בעצמכם והסבירו בחירותיכם בדו"ח.

4. יבאו `DataCollatorForLanguageModeling` מתוך transformers וצרו Trainer מתאים למשימה:

```
from transformers import DataCollatorForLanguageModeling
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_dataset,
    data_collator=DataCollatorForLanguageModeling(tokenizer=tokenizer,
mlm=False),)
```

5. אמנו את המודל:

```
trainer.train()
```

6. שמרו את המודל ואת הtokenizer לשימוש עתידי:

```
trainer.model.save_pretrained(save_directory)
tokenizer.save_pretrained(save_directory)
```

7. טענו את המודל:

```
model = GPT2LMHeadModel.from_pretrained(save_directory)
tokenizer = GPT2Tokenizer.from_pretrained(save_directory)
```

8. צרו input_ids ומסיכת attention עבור "The movie was"=prompt:

```
input_ids = tokenizer.encode(prompt, return_tensors="pt")
attention_mask = input_ids.ne(tokenizer.pad_token_id)
```

9. ג'נרטו ביקורת בעזרת כל אחד מהמודלים שאימנתם :

```
with torch.no_grad():
    output = model.generate(
        input_ids,
        attention_mask=attention_mask,
        max_length=max_length,
        temperature=temperature,
        top_k=top_k,
        top_p=top_p,
        repetition_penalty=repetition_penalty,
        do_sample=True,
        num_return_sequences=1
    )
generated_text = tokenizer.decode(output[0], skip_special_tokens=True))
```

בחרו את הפרמטרים בעצמכם והסבירו בחירותיכם בדו"ח.

10. הדפיסו לקובץ בשם generated_reviews.txt 5 ביקורות עם המודל החיובי ו5 עם המודלי השלילי בפורמט הבא:

Reviews generated by positive model:

1. <first generated review>
2. <second generated review>

   <and so on ....>

Reviews generated by negative model:

1. <first generated review>
2. <second generated review>

   <and so on ....>

הערות:

1. השתמשו ב100 ביקורות בלבד לכל אחד מסטי האימון, על מנת להגביל את זמן אימון המודל.

ענו על השאלות הבאות:

1. האם פלטי המודלים תאמו לציפיות שלכם? הסבירו.
2. האם ראיתם הבדלים משמעותיים בתוצרים של כל אחד מהמודלים? פרטו.
3. הסבירו מה היה משתנה אילו היינו מגדילים ואילו היינו מקטינים באופן משמעותי את כמות הדוגמאות בסטי האימון. התייחסו הן לתוצאות והן לתהליך האימון.
4. הסבירו מה התפקיד של הattention mask שיצרתם בסעיף 8.
5. עד כמה לדעתכם הprompt שהעברנו למודל משמעותי עבור התוצאה? התנסו בprompts אחרים לבחריתכם ובדקו את השערתכם. פרטו בדו"ח.

חלק ד': Prompt Engineering

הקדמה: הנדסת פרומטים היא תהליך עיצוב ושיפור הפרומפטים (הנחיות הטקסט המהוות קלט למודל) שאנו מספקים למודלי שפה גדולים (LLMs) מודרנים כמו GPT-4, כדי להנחות אותם לבצע משימה מסויימת בצורה מדוייקת ויעילה, ולקבל מהם תשובות רצויות.

מודלים אלה מאומנים על מאגרי נתונים גדולים והם מסוגלים לבצע משימות רבות מבלי להידרש לאימון מחדש. עם זאת, התשובה שהם נותנים תלויה באיכות ובמבנה של הפרומפט (ההנחיה) שאנחנו מספקים להם. הנדסת פרומפטים מאפשרת לנו להשפיע על הפלט של המודל ולהשיג תוצאות מדוייקות וממוקדות יותר, על ידי עיצוב נכון של הפרומפטים.

למה צריך את זה?

- **מיקוד המשימה:** פרומפטים טובים מגדירים באופן ברור מהי המשימה שעל המודל לבצע, ומפחיתים את הסיכוי לפלט לא רלוונטי. הנדסת פרומפטים יכולה לשפר את הביצועים של המודל על משימות כמו תרגום, סיווג, מענה לשאלות ועוד.
- **חיסכון בזמן:** במקום לבצע Fine-Tuning (אימון נוסף) למודל על מאגר נתונים חדש, ניתן להשתמש בהנדסת פרומפטים כדי להפיק את התשובות הרצויות ללא צורך בעדכון הפרמטרים של המודל.
- **למידה מועטה: (Few-shot learning)** הנדסת פרומפטים מאפשרת למודל לבצע משימות גם כשיש מעט מאוד דוגמאות (Few-shot) או אפילו ללא דוגמאות כלל (Zero-shot) . במקרים אלו, ניתן להשתמש בפרומפטים מתוכננים היטב כדי לכוון את המודל.

אסטרטגיות להנדסת פרומפטים:

1. Zero-shot Prompting:

בפרומפט Zero-shot, אנו מבקשים מהמודל לבצע את המשימה מבלי לספק לו דוגמאות קודמות.
לדוגמה, אם נרצה שהמודל יתרגם משפט מאנגלית לעברית, נוכל להציע לו פרומפט כמו:

"Translate the following sentence from English to Hebrew: 'The cat is sitting on the mat'"

בפרומפט זה, המודל מבצע את המשימה על סמך המידע שהוא למד במהלך האימון, ללא דוגמאות נוספות.

2. Few-shot Prompting:

בפרומפט מסוג זה אנו מספקים למודל מספר דוגמאות לפני שאנו מבקשים ממנו לבצע את המשימה.
לדוגמה, נספק למודל מספר דוגמאות של משפטים מתורגמים, לפני שאנו מבקשים ממנו לתרגם משפט חדש. זה עוזר למודל להבין בצורה ברורה יותר את המשימה:

"Translate the following sentences from English to Hebrew:
1. ***'The sun is shining.' : 'השמש זורחת.'***
2. ***'The dog is barking' : 'הכלב נובח.'***

Now translate this sentence: 'The cat is sitting on the mat'. "

2. Instruction-based Prompting:

בפרומפט מסוג זה, אנו מספקים הוראות ברורות ומפורטות כדי לכוון את המודל איך לבצע את המשימה. לדוגמה:

"You are a translator. Please translate the following English sentence to Hebrew: 'The cat is sitting on the mat.' Make sure the translation is accurate and natural".

עליכם לבצע סיווג רגשות על מאגר IMDb באמצעות שלושת הטכניקות להנדסת פרומפטים. נשתמש במודל flan-T5 שהינו encoder-decoder.
לשם כך:

1. טענו את המודל flan-T5 ואת הtokenizer המתאים לו:

```
tokenizer = AutoTokenizer.from_pretrained("google/flan-t5-small")
model = AutoModelForSeq2SeqLM.from_pretrained("google/flan-t5-small")
```

2. דגמו באופן אקראי 50 reviews מתוך הIMDB, בצורה stratified, כלומר עם מספר דוגמאות חיוביות ושליליות שוות.
3. צרו פרומפטים מהסוגים הבאים:
   - **Zero-shot Prompting**: צרו פרומפט שמבקש מהמודל לסווג ביקורות סרטים כחיוביות או שליליות מבלי לספק דוגמאות.
   - **Few-shot Prompting**: צרו פרומפט שבו אתם מספקים 2 דוגמאות – אחת של ביקורת חיובית ואחת של שלילית, לפני שאתם מבקשים מהמודל לסווג ביקורת חדשה.
   - **Instruction-based Prompting**: צרו פרומפט מפורש שמדריך את המודל לבצע את המשימה בצורה מדויקת.
4. סווגו בעזרת המודל את הreviews שדגמתם בסעיף 2 בעזרת כל אחת מאסטרטגיות הפרומפט.
   הדפיסו לקובץ בשם flan_t5_imdb_results.txt את התוצאות בפורמט הבא:

   Review 1: <first sample from sampled IMDB reviews>
   Review 1 true label: <true label from IMDB>
   Review 1 zero-shot: <output for this review with the zero-shot prompt>
   Review 1 few-shot: <output for this review with the few-shot prompt>
   Review 1 instruction-based: <output for this review with the instruction-based prompt>
   Review 2: <second sample from sampled IMDB reviews>
   …. <and so on>

Review 50: <…>

5. דווחו על ביצועי הaccuracy של כל אסטרטגיית פרומפט ונתחו את איכות התוצאות.
6. צרפו לדו"ח את הpromptsשיצרתם ודוגמאות לסיווגים מוצלחים ולא מוצלחים של כל אחת מהשיטות.

<u>הערות</u>:
1. הפלט שעליכם לבקש מהמודל להחזיר הוא "positive" לביקורת חיובית ו"negative" לשלילית.
2. התעלמו מהבדלי case sensitive ורווחים בתשובות. כלומר "**P**ositive" ו "**p**ositive" היא אותה תוצאה.
3. פלט שלא עומד בתנאים שביקשנו (כלומר לא החזיר positive/negative (או Positive/Negative) אלא מחרוזת אחרת, ייחשב כסיווג לא נכון.

ענו על השאלות הבאות:
1. האם התוצאות שקיבלתם תאמו לציפיות שלכם? הסבירו.
2. האם התוצאות יהיו שונות אם תתנו יותר מ2 דוגמאות עבור הfew-shot prompt? הסבירו.
3. הסבירו את ההבדל בין Fine-Tuning מסורתי לבין למידה מבוססת פרומפטים (prompt-based learning). במה הם שונים ובמה הם דומים?
4. האם תמיד אפשר להחליף fine-tuning בprompt-based learning?
5. כפי שראיתם, ניסוח הפרומפט יכול להשפיע משמעותית על אופי ואיכות התוצאה. דבר זה מקשה מאוד על האיבלואציה של המודלים – עד כמה הם מצליחים במשימה מסויימת, ובהשוואה בין מודלים- בדיקה לאיזה מודל יש ביצועים טובים יותר על אותה משימה. הסבירו מדוע זה מקשה על האיבלואציה והציעו פתרונות אפשריים.

חלק ה': Bias

לתרגיל מצורפת מצגת המציגה דוגמה לbias באפליקציה של ChatGPT.
ענו על השאלות הבאות:

1. התמונות שהמודל הפיק לא תאמו לבקשת המשתמשת. האם אתם מסכימים עם הקביעה שמדובר בbias? הסבירו.
2. מה, ככל הנראה, הסיבות שגרמו לו להפיק את התוצאות הנ"ל?
3. נסו בעצמכם! בקשו מ chatGPT לייצר תמונה של אישה נוהגת באוטו וגבר לידה (מוזמנים להשתמש בכל פרומפט משלכם לשם כך). האם קיבלתם את התוצאה הרצויה או שגם אתם נתקלתם באותה בעיה? צרפו את הפרומפט שלכם והתוצאה שקיבלתם לדו"ח.

4. **שאלת בונוס:** האם תצליחו למצוא bias נוסף במודל? בחרו באחד מהצ'אט-בוטים מבוססי LLMs הידועים (ChatGPT, Claude, Gemini...) והראו סיטואציה שבה התוצאה שהם מפיקים נגועה בbias.
(השאירו סעיף זה לסוף ונסו לעשות אותו <u>רק אם נותר לכם זמן</u>, על מנת לא לבזבז זמן יקר).

<u>הערה על התרגיל:</u>
אנחנו לא נגענו בקורס בשימוש בGPU ובנושא של CUDA. אני מניחה שרובכם תלמדו על כך בקורס למידה עמוקה. כמו כן, אני לא יודעת איזה סוג חומרה יש לכם במחשבים האישיים שלכם. לכן, אין דרישה להשתמש בGPU בתרגיל זה ומספר הדוגמאות שעליכם להשתמש בהם לאורך התרגיל הוא קטן יחסית (ואף קטן מדי), על מנת לאפשר זמני ריצה סבירים גם על הCPU. עם זאת, למי שיש אפשרות, אני כמובן ממליצה להריץ את הכל עם GPU על מנת להקל עליכם.

ספריות מותרות לשימוש

אתם יכולים להשתמש ב

Pandas, Numpy, scikit-learn, torch, transformers, datasets

ובכל ספריה סטנדרטית של python.

אתם יכולים לחפש שם של ספריה ב https://docs.python.org/3/library/index.html על מנת לבדוק אם זו ספריה סטנדרטית. לא יהיה מענה על שאלות לגבי שימוש בספריות ספציפיות.

- למען הסר ספק, json היא ספרייה סטנדרטית של python.
- מומלץ להשתמש עבור כל פרוייקט בסביבה וירטואלית virtual environment חדשה משלו על מנת להיות בטוחים שאתם משתמשים רק בספריות מותרות ולמנוע קונפליקטים עם ספריות קודמות שהתקנתם בעבר. ראו מצגת על כך במודל.

הערות כלליות

1. על הקוד שלכם להיות מסוגל להתמודד עם שגיאות בכל שלב בתהליך ולא לקרוס. השתמשו ב-Try Except blocks לפי הצורך.
2. שימו לב, בבדיקת תרגילי הבית בקורס ניתן משקל גדול מהניקוד הן על **הדו"ח**, ההסברים והידע שהפגנתם בחומר הנלמד והן על **הקוד**, אופן המימוש, יעילותו, קריאותו ועמידותו. בפרט, הרבה מהבדיקות הן אוטומטיות ולכן עליכם להקפיד על קוד תקין שרץ ללא שגיאות ועל עמידה **מדוייקת** בפלט הנדרש וביתר ההנחיות.
3. ניתן לשאול שאלות על התרגיל בפורום המיועד במודל. למעט מקרים אישיים מיוחדים, אין לשלוח שאלות הקשורות לתרגיל הבית במייל.
4. על אחריותכם לעקוב אחר הודעות הקורס במודל (בלוח הודעות ובפורום) ולהיות מעודכנים במידה ויהיו שינויים בהנחיות.

אופן ההגשה

1. ההגשה היא בזוגות בלבד.
2. עליכם להגיש קובץ zip בשם hw5_<id1>_<id2>.zip (כאשר <id1>,<id2> הם מספרי תעודות הזהות של הסטודנט הראשון והשני בהתאמה), המכיל את הקבצים הבאים:
   a. קובץ python בשם **bert_classification_finetuning.py** המכיל את כל הקוד הנדרש כדי לממש את חלק ב'.
      i. הקלט לקובץ יהיה הנתיב למיקום של הIMDB subset.
      ii. הפלט יהיה הדפסה למסך של הaccuracy כפי שתואר בחלק ב'.

      על הקובץ לרוץ תחת הפקודה הבאה:

```
python bert_classification_finetuning.py <path/to/imdb subset>
```

   b. קובץ python בשם **gpt_generation_finetuning.py** המכיל את כל הקוד הנדרש כדי לממש את חלק ג'.

i. הקלט לקובץ יהיה הנתיב למיקום של הIMDB subset , נתיב לתיקייה עבור שמירת המודלים, נתיב לקובץ הפלט.

ii. הפלט יהיה קובץ הgenerated_reviews.txt כפי שהוסבר בחלק ג' והמודלים שאומנו שישמרו בתיקייה שהתקבלה בקלט.

על הקובץ לרוץ תחת הפקודה הבאה:

python gpt_generation_finetuning.py <path/to/imdb subset> <path/to/generated_reviews.txt> <path/to/saved models dir>

c. קובץ Python בשם **flan_t5_prompt_engineering.py** המכיל את כל הקוד הנדרש כדי לממש את חלק ד'.

i. הקלט לקובץ יהיה נתיב למיקום של הIMDB subset, נתיב לקובץ הפלט flan_t5_imdb_results.txt כפי שהוסבר בחלק ד'.

על הקובץ לרוץ תחת הפקודה הבאה:

python flan_t5_prompt_engineering.py <path/to/imdb subset> <path/to/flan_t5_imdb_results.txt>

d. קובץ טקסט השם **generated_reviews.txt** כפי שתואר בחלק ג'.

e. קובץ טקסט בשם **flan_t5_imdb_results.txt** כפי שתואר בחלק ד'.

f. קובץ PDF בשם **<id1>_<id2>_hw5_report.pdf** ובו דו"ח המפרט על הקוד, על ההחלטות שקיבלתם במהלך העבודה על התרגיל ומענה על כל השאלות בכל החלקים.
<u>אל תשכחו לציין בתחילת הדו"ח את שמותיכם **בעברית** ותעודות הזהות שלכם.</u>

יש להקפיד על עבודה עצמית, צוות הקורס יתייחס בחומרה להעתקות או שיתופי קוד, כמו גם שימוש בכלי AI דוגמת chatGPT.

ניתן לשאול שאלות על התרגיל בפורום הייעודי לכך במודל.

יש להגיש את התרגיל עד לתאריך 30.01.25 בשעה 23:59.

בהצלחה!