מכון טכנולוגי חולון
Holon Institute of Technology

51095 מודלי יסוד: ארכיטקטורה, אימון ויישור

מעבדה יישומית 1 - Inside a Foundation Model

מטרת המעבדה: להבין באופן מעשי כיצד טקסט נכנס למודל שפה, כיצד הוא מומר לטוקנים, כיצד נוצרים ייצוגים וקטוריים, וכיצד המודל מייצר המשך טקסט באמצעות חיזוי הטוקן הבא.

1. מטרות למידה

- להסביר מדוע מודל שפה אינו מקבל מילים אלא טוקנים.
- להדגים כיצד טוקנייזרים שונים מחלקים טקסט באופן שונה.
- להבין כיצד משפטים מיוצגים כווקטורים באמצעות Embeddings.
- לחשב דמיון סמנטי בין משפטים באמצעות Cosine Similarity.
- להמחיש ברמה אינטואיטיבית מהו Attention.
- להבין את ההבדל בין מודלים מסוג BERT לבין מודלים מסוג GPT.

2. רקע קצר

מודל שפה מודרני אינו עובד ישירות עם טקסט כפי שבני אדם קוראים אותו. לפני שהטקסט נכנס למודל, הוא עובר תהליך טוקניזציה: חלוקה ליחידות קטנות יותר הנקראות טוקנים. כל טוקן מומר למספר, והמספרים מומרים לייצוגים וקטוריים. בתוך ה-Transformer, מנגנון ה-Attention מאפשר לכל טוקן להתייחס לטוקנים אחרים בהקשר.

המעבדה אינה נועדה לממש Transformer מאפס, אלא לפתוח את הקופסה השחורה ברמה אינטואיטיבית ומעשית.

3. מבנה המעבדה

חלק	נושא	תוצר
א	Tokenization	השוואת טוקנים וטוקן IDs עבור משפטים שונים
ב	Embeddings	ייצוג משפטים כווקטורים וחישוב דמיון סמנטי
ג	Attention	הצגת Heatmap של Attention Head
ד	GPT vs BERT	השוואה מושגית והתנסות קצרה
ה	Reflection	תשובות קצרות על משמעות הממצאים

מכון טכנולוגי חולון
Holon Institute of Technology

51095 מודלי יסוד: ארכיטקטורה, אימון ויישור

מעבדה יישומית 1 - Inside a Foundation Model

4. סביבת עבודה

יש לפתוח את מחברת ה-Colab המצורפת ולהריץ את התאים לפי הסדר. ברוב החלקים אין צורך ב-GPU, אך מומלץ להפעיל GPU אם הוא זמין כדי לקצר זמני טעינה.

5. חלק א: Tokenization

בחלק זה נבחן כיצד טקסט רגיל הופך לרשימת טוקנים ולרשימת מספרים. שימו לב במיוחד למילים ארוכות, לטקסט בעברית, למספרים, לכתובות URL ולאימוג׳י.

```
from transformers import AutoTokenizer

tokenizer = AutoTokenizer.from_pretrained("bert-base-uncased")

texts = [
    "Transformers are amazing!",
    "internationalization",
    "The price is 12,500 NIS",
    "https://www.example.com",
    "שלום עולם",
    "AI is 🔥"
]

for text in texts:
    print("TEXT:", text)
    print("TOKENS:", tokenizer.tokenize(text))
    print("IDS:", tokenizer.encode(text))
    print("-")
```

שאלות קצרות

1. איזה טקסט התפרק למספר הטוקנים הגדול ביותר?
2. מה קרה לטקסט בעברית?
3. מדוע לדעתכם Tokenization משפיע על עלות השימוש במודל?

6. חלק ב: Embeddings ודמיון סמנטי

בחלק זה נשתמש במודל Sentence Transformer כדי להפוך משפטים לווקטורים. לאחר מכן נחשב דמיון סמנטי בין שאילתה לבין אוסף משפטים.

```
from sentence_transformers import SentenceTransformer, util

model = SentenceTransformer("all-MiniLM-L6-v2")

sentences = [
    "A dog is running in the park.",
```

```
    "A puppy is playing outside.",
    "The stock market crashed yesterday.",
    "Investors are worried about inflation.",
    "I ate pizza for dinner."
]

query = "A small dog is playing."

sentence_embeddings = model.encode(sentences)
query_embedding = model.encode(query)

scores = util.cos_sim(query_embedding, sentence_embeddings)[0]

for sentence, score in sorted(zip(sentences, scores), key=lambda x: x[1], reverse=True):
    print(f"{score:.3f} | {sentence}")
```

החליפו את המשפטים בדוגמאות מתחום שמעניין אתכם: רפואה, משפטים, פיננסים, חינוך או הנדסת תוכנה.

7. חלק ג: המחשת Attention

בחלק זה נטען מודל BERT ונבקש ממנו להחזיר את מטריצות ה-Attention. לאחר מכן נציג Heatmap עבור שכבה וראש Attention אחדים.

```
import matplotlib.pyplot as plt
from transformers import AutoTokenizer, AutoModel

model_name = "bert-base-uncased"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModel.from_pretrained(model_name, output_attentions=True)

text = "The animal did not cross the street because it was too tired."
inputs = tokenizer(text, return_tensors="pt")
outputs = model(**inputs)

tokens = tokenizer.convert_ids_to_tokens(inputs["input_ids"][0])
attentions = outputs.attentions

layer = -1
head = 0
attention = attentions[layer][0, head].detach().numpy()

plt.figure(figsize=(9, 7))
plt.imshow(attention)
plt.xticks(range(len(tokens)), tokens, rotation=90)
plt.yticks(range(len(tokens)), tokens)
plt.colorbar()
plt.title(f"Attention heatmap - layer {layer}, head {head}")
plt.show()
```

מכון טכנולוגי חולון
Holon Institute of Technology

51095 מודלי יסוד: ארכיטקטורה, אימון ויישור

מעבדה יישומית 1 - Inside a Foundation Model

הערה: Attention הוא כלי חשוב להבנת הארכיטקטורה, אך אין לראות בו הסבר מלא להחלטות המודל. זהו חלון חלקי בלבד אל אופן חישוב הייצוגים הפנימיים.

8. חלק ד: GPT מול BERT

מאפיין	BERT	GPT
כיוון	דו-כיווני / Encoder	חד-כיווני / Decoder
משימה טיפוסית	הבנת טקסט, סיווג, ייצוגים	השלמת טקסט ויצירת טקסט
אימון בסיסי	Masked Language Modeling	Next Token Prediction
שימוש מרכזי	Embedding והבנת הקשר	Generation ושיחה

במחברת המצורפת ניתן להריץ דוגמת Text Generation קצרה עם GPT-2 כדי להמחיש את רעיון חיזוי הטוקן הבא.

9. תרגיל עצמאי

בחרו תחום תוכן אחד שמעניין אתכם ובנו אוסף קטן של עשרה משפטים או פסקאות קצרות. לאחר מכן בצעו:

- טוקניזציה של שלושה משפטים לפחות.
- חישוב Embeddings לכל המשפטים.
- שאילתה אחת שמחזירה את שלושת המשפטים הדומים ביותר.
- הסבר קצר: האם התוצאה הגיונית? אם לא, מדוע?

10. אתגר למתקדמים

בנו מנוע חיפוש סמנטי קטן: המשתמש מזין שאילתה, והקוד מחזיר את שלושת המסמכים הדומים ביותר מתוך אוסף מסמכים קצר. זהו הבסיס האינטואיטיבי למערכות RAG שנלמד בהמשך הקורס.

```
def semantic_search(query, documents, model, top_k=3):
    doc_embeddings = model.encode(documents)
    query_embedding = model.encode(query)
    scores = util.cos_sim(query_embedding, doc_embeddings)[0]
    ranked = sorted(zip(documents, scores), key=lambda x: x[1], reverse=True)
    return ranked[:top_k]

results = semantic_search("How do language models represent meaning?", sentences, model)
for doc, score in results:
    print(f"{score:.3f} | {doc}")
```

מכון טכנולוגי חולון
Holon Institute of Technology

51095 מודלי יסוד: ארכיטקטורה, אימון ויישור

מעבדה יישומית 1 - Inside a Foundation Model

11. שאלות רפלקציה להגשה

1. מה הפתיע אתכם בתהליך הטוקניזציה?
2. האם דמיון סמנטי הוא בהכרח אמת עובדתית? הסבירו.
3. מה היתרון של Embeddings לעומת חיפוש לפי מילות מפתח?
4. מה ההבדל המרכזי בין מודל שמייצר טקסט לבין מודל שמייצג טקסט?
5. כיצד המעבדה הזו מתחברת למערכות RAG שנבנה בהמשך?

12. הגשה קריטריוני בדיקה

יש להגיש את המעבדה כקובץ ג'ופיטר יחיד. את התוצאות וההסברים (התמציתיים מאוד) יש לכתוב בתוך המחברת.

- הרצת כל תאי המחברת - חובה.
- דוגמאות מקוריות בתחום שנבחר - חובה.
- תשובות לשאלות הרפלקציה - חובה.
- Semantic Search עצמאי - מומלץ / אתגר.
- הסברים ברורים ומדויקים - חובה.

13. סיכום

במעבדה זו פתחנו את צינור העיבוד הבסיסי של מודל יסוד: טקסט נכנס, עובר Tokenization, מיוצג באמצעות Embeddings, מעובד באמצעות Attention, ולבסוף יכול לשמש להבנת טקסט או ליצירת טקסט. בשיעורים הבאים נעמיק בארכיטקטורות, אימון, התאמה, RAG, Agents ו-Evaluation.