

סיכום שיעור

חלונות הקשר ו-RAG

Context Windows and RAG

ד"ר יורם סגל

כל הזכויות שמורות לד"ר יורם סגל (C)

מילות מפתח

- חלון הקשר -- Context Window
- יצירה מוגברת באחזר -- RAG -- Retrieval-Augmented Generation
- מודלי שפה גדולים -- LLM -- Large Language Models
- טוקנים -- Tokens
- וקטורים -- Vectors
- צ'אנקים -- Chunks
- חיפוש סמנטי -- Semantic Search
- מסד נתונים וקטורי -- Vector Database
- עקומות ה-U -- Lost in the Middle
- הנחיתת מערכת -- System Prompt

1 חלון הקשר -- המודל והאנלוגיה

1.1 הבנת המודל הבסיסי

מודלי שפה גדולים (LLMs) כמו GPT ו-Claude הם חסרי מצב (stateless) -- כל שיחה מתחילה מאפס ואין להם זיכרון הקשורות קודמות. הפתרון הוא חלון הקשר (Context Window), שמאפשר לנו להנедיר את המידע בכל שיחה.

2.1 האנלוגיה למחשב

ניתן להבין את המערכת באמצעות אנלוגיה למחשב:

- ה-LLM שקול ל-CPU -- המעבד שמבצע את החישובים
- חלון הקשר שקול ל-RAM -- האזיכון הזמן
- מסד הנתונים הוקטורי שקול ל-Hard Disk -- אחסון קבוע
- המשתמש הוא מערכת הפעלה -- ניהול את זרימת המידע

חשוב להבין את היררכיית המהירות: רגיסטרים הם המהירים ביותר (ויקרים ביותר), אחריהם Cache, RAM, ולבסוף Hard Disk (איטי אך זול). ההבדל ב מהירות בין RAM ל-Hard Disk הוא פי אלף -- כמו ההבדל בין טיסה להודו לבין הליכה רגל.

3.1 טוקנים ווקטורים

כל טקסט מומר לטוקנים, וכל טוקן מיוצג כווקטור. כללי אצבע חשובים:

- באנגלית: כל טוקן מכסה כ- 0.7 מיליה
- בעברית: כל מילה צריכה כ- 2.5 טוקנים (בגלל שימוש נמוך יחסית בעברית באימון)
- הווקטור מורכב מkomponenotot שכלל אחת מייצגת מאפיין סמנטי

4.1 עיקומת ה-U -- אובדן מידע באמצעות

מחקר פורץ דרך מ-2023 גילתה תופעה קריטית: מודלי שפה שוכחים מידע שנמצא באמצעות חלון ההקשר. הביצועים במצבה מידע:

- בהתחלה -- גבויים מאוד (Attention חזק)
- באמצעות -- צונחים ל-50%
- בסוף -- חוזרים לרמה גבוהה

המסקנה המעשית: אל תשים מידע קריטי באמצעותו! מידע חשוב צריך להופיע בתחילת ההקשר או בסופו.

5.1 אסטרטגיות ניהול חלון ההקשר

ארבע אסטרטגיות מרכזיות:

1. **כתיבה תמציתית (Write)** -- שימוש בתבניות ומיעור טקסט מיותר, קבצים קטנים (עד 150 שורות)
2. **סיכום תקופתי (Select)** -- סיכום חלון ההקשר לאחר כל שאלה
3. **אחיזור סלקטיבי (Compress)** -- שימוש בחיפוש סמנטי לאחיזור המסמכים הרלוונטיים ביותר
4. **בידוד (Isolate)** -- שמירה על חלונות הקשר נפרדים לסוכנים או משימות שונות

6.1 הכללה של חלון ההקשר

יש הבדלים משמעותיים במחיר בין מודלים:

- 128,000 טוקנים, \$0.15 למיליאון (הבסיס)
- 200,000 טוקנים, \$3.00 למיליאון (פי 20 יקר יותר)
- 128,000 טוקנים, \$2.50 למיליאון -- GPT-4o
- 2,000,000 טוקנים, \$1.25 למיליאון -- Gemini 1.5 Pro

פתחים ומנהלים חייבים לש拷ול את האיזון בין ביצועים, גודל חלון ההקשר וועלויות.

2 -- יצירה מוגברת באחזר RAG

1.2 הבעה היסודית

מודלי שפה גדולים הם כמו גאנונים עם זיכרונות מושלים, אך עם מגבלות:

-- **סתטיים** -- הידע שלהם כפוא בזמן האימון

-- **כלליים** -- אין להם גישה למידה ספציפי של הארגון

-- **מוגבלים** -- לא יכולים לקרוא מיליוני מסמכים בbase אחת

2.2 הפתרון -- הספרון הדיגיטלי

(Retrieval-Augmented Generation) RAG היא טכניקה שהופכת את המודל לחכם יותר:

1. **שאילתת** -- המשמש שאל שאלת

2. **חיפוש** -- המערכת מחפשת במאגר מסמכים חיצוני

3. **אחזר** -- שליפת המסמכים הרלוונטיים ביותר (Top-K)

4. **בנייה של קשר** -- הוספת המסמכים לchl Lon ההקשר

5. **יצירת תשובה** -- המודל עונה על בסיס המידע שהוחזר

3.2 ארכיטקטורת RAG

המערכת מורכבת מארבעה רכיבים:

-- **Query Encoder** -- מקודד את השאלה לווקטור

-- **Document Encoder** -- מקודד מסמכים לווקטורים (חדר פעמי)

-- **מנוע אחזר** -- מחשב מרחק וקטורי (בדרכן כלל קוסינוס)

-- **Generator** -- מייצר תשובה על בסיס המסמכים

חשוב: Document Encoder ו-Query Encoder מודלים

4.2 שתי גישות ל-RAG

1. **Sequence RAG** -- אחזר בתחלת התהליך, יוצרת תשובה שלמה

2. **Dynamic RAG** -- אחזר מרובד במהלך ייצור התשובה (כמו Deep Research)

5.2 צ'אנקינג -- פיצול מסמכים

שלוש שיטות עיקריות:

-- **לפי טוקנים** -- כל 512 טוקנים עם חפיפה של 50. פשוט אך עלול לחזור באמצעות רעיון

-- **לפי משפטים** -- צירוף משפטים עד 400 טוקנים. שומר על שלמות

-- **סמנטי** -- פיצול לפי נושאים עם Embedding. מורכב אך מדויק

6.2 מסדי נתונים וקטוריים

המהפכה של 2019 (Sentence-BERT) אפשרה ליצור Embedding עצמאי לכל מסמך ולשמור אותו. מסדי נתונים וקטוריים כמו Chroma מאפשרים מהיר פי אלף.

7.2 דוגמאות מעשיות

-- יצירת סוכן עם בסיס ידע ב-GPTs --

-- הוספת מסמכים לפרויקט ב-Claude Projects --

-- מנוע חיפוש שמסכם אתרים ב-Perplexity --

8.2 בעיות נפוצות ב-RAG

-- אין מידע -- המידע לא קיים במאגר

-- Not in Top-K -- המסמך הרלוונטי לא בתוצאות הראשונות

-- Lost in the Middle -- המידע באמצע ולא נמצא

-- Not Extracted -- המודל לא חילץ נכון (בעיקר בטבלאות)

-- שרשרת פישלון -- כל שלב תלוי בקודמו

פתרון מומלץ: השתמשו ב-Few-Shot Examples! תנו דוגמאות של קלט ופלט רצויים.

3 הזכות לשכוח -- היבט פילוסופי

1.3 חלון ההקשר המוגבל כפייצ'ר

חלון ההקשר המוגבל אינו באג -- הוא פיצ'ר. השכחה מאפשרת:

-- שחרור מה עבר -- סוכן שטעה לא סוחב את הטיעות אלף צעדים קדימה

-- גמישות ויעילות -- לא כל מידע חשוב לנצח

-- התחלת מחדש -- כמו הזכות לשכוח במשפט נגד גוגל

2.3 מה מבידיל אותנו מ-AI?

-- רגש אמיתי -- לסוכן יש רק סטטיסטיקה של רגש

-- הזכות לטיעות בכוונה -- אנחנו יכולים לבחור ללקת נגד הוויז'

-- תשומת לב מוגבלת -- אנחנו בוחרים במה להתמקד

-- חוויה -- אנחנו נהנים וסובלים, לא רק מעבדים

3.3 כלכלת AI חיובית ושלילית

תמיד חייב להיות אדם בLOOP (Human-in-the-Loop) שיכל:

- להוריד את המתג
- לקבע סדר עדיפויות
- לשמור על מיקוד ולא להיסחף

4 המטלה -- ניסוי מחקרי'

1.4 גישה מחקרית

המטלה הפעם שונה -- לא תכנות אלא מחקר:

- נשחו שאלת מחקר
- הציבו השערה
- תכננו ניסוי
- הצינו ממצאים בצורה יצירתיות

2.4 נושאי מחקר מוצעים

1. **Lost in the Middle** -- בדיקת ירידה בדיק כשמיידע באמצעות

2. **השפעת גודל** -- כיצד גודל חלון ההקשר משפיע על ביצועים

3. **אסטרטגיות ניהול** -- השוואת Write>Select/Compress/Isolate

4. **הצברות בעיות** -- כיצד בעיות מצטברות בסוכנים