המאמר היומי של מייק: 20.05.25  
Don’t Do RAG: When Cache-Augmented Generation is All You Need for Knowledge Tasks

3 ימים בלי סקירה עקב עומס מטורף ואירועים משפחתיים לא מאוד משמחים. אז בחרתי מאמר ממש קליל שאפשר לסקור אותו בכמה משפטים בודדים. בנוסף הוא קשור לראג ומזמן לא סקרתי מאמר על הנושא הפופולרי הזה. אוקיי, מה זה ראג? זה בעצם מודל שפה המצויד בזיכרון חיצוני המשלים את הידע של LLM שעוזר לו (מודל שפה) לבנות תשובה יחד עם הידע מהזיכרון. כלומר במצב אידיאלי אנו רוצים לשלב את הידע שנצבר במודל שלנו יחד עם הזכרון (נגיד אוסף מסמכים) כדי לקבל תשובה אופטימלית.

אז איך מוצאים בראג מסמכים רלוונטיים? בפשטות לפי דמיון בין האמבדינגס של המסמך והאמבדינג של שאלה - בוחרים אז מה החיסרון הכי גדול של ראג? צריך לשמור את כל האמבדינג האלו הזכרון מהיר כדי שהמודל ימצא את המסמכים הרלוונטיים ויג'נרט לנו את התשובה מהר. מכיוון שיש לנו לפעמים מיליוני מסמכים זה יכול להיות די יקר למרות שיטות חיפוש מאוד יעילות הקיימות היום (vector database).

אז המחברים הציעו שיטה די אינטואיטיבית לחיסכון בזיכרון של ראג באמצעות שימוש בקאש. אנו נשמור את המסמכים בזיכרון של המודל כמו kv-cache. פשוט נדחף בחלון הקונטקסט של המודל את כל המסמכים ונחשב את ה kv-cache עבור כל הטוקנים של כל המסמכים. מחשבים את ה kv-cache הזה מראש וזה מייתר לנו את הצורך בהשוואת האמבדינגס של השאלה ושל המסמכים. ואז דוחפים את השאלה מודל שהוא prefill עם המסמכים האלו ועושים אינפרנס עבורה. המאמר רומז שדוחפים למודל את השאלות הקודמות עד שנגמר חלון ההקשר (לא בטוח שאני מבין למה).

אז כמובן שיש לשיטה הזו מגבלות - סט מסמכים צריכים להיות קטן מספיק כדי להיכנס לחלון ההקשר של המודל (למרות שבמודלים החדשים שכביכול יש חלון הקונטקסט די ארוך). ועוד משהו קטן: הרי סדר של מסמכים לא רלוונטי שמאלץ אותנו לסמוך על חכמת המודל שיידע להתעלם מזה ולהשתמש בדאטה נחוץ בלבד.

מאמר די טריוויאלי אבל צריך לסקור גם כאלו לפעמים

https://arxiv.org/pdf/2412.15605