⚡️🚀המאמר היומי של מייק 09.09.24: ⚡️🚀

MemLong: Memory-Augmented Retrieval for Long Text Modeling

אחד המאמרים ראשוניים בנושא Retrieval Augmented Generation או RAG שאני סוקר. הנושא צובר תאוצה רצינית בזמן האחרון והגיע הזמן להשלים את הפערים (גם בידע וגם בסקירות).

RAG זה בעצם דרך להתגבר על כך שלמרות כל ההישגים בתחום אפילו מודלי שפה החדשים ביותר מתקשים לעבוד עם אורך הקשר מאוד ארוך. מה בעצם קורה כאן? נניח שיש לנו דאטהסט D ואנחנו רוצים שמודל השפה שלנו יענה על שאלות על D תוך כדי שילוב יכולות שהוא צבר במהלך האימון לפני זה.

אחת הדרכים היא לעשות למודל שפה פיינטיון על D אולם זה עלול להיות בעייתי כי המודל יכול לשכוח חלק מהדברים שידע קודם וגם יתקשה ללמוד את כל מה שיש ב-D בצורה יעילה (פתיר כמובן אבל קשה). הדרך השנייה כי להוסיף את D לכל שאלת המשתמש (כחלק מפרומפט) אבל זה גם בעייתי ל- D גדולים עקב אי יכולת של מודלי שפה להתמודד עם אורך הקשר גדול מאוד.

דרך נוספת היא לעשות RAG (אפשר לשלב אותו עם פיינטיון קליל - ראיתי מאמר שעושה את זה) כלומר לכל שאילתה של משתמש לבחור את המידע מ- D (כמה צ'אנקים) הכי רלוונטיים לשאלה והוסיף אותם לפרומפט. הבעיה בגישה הזו היא מטריקה לבחירת הצ'אנקים הרלוונטיים ביותר לשאלה. בד״כ זה נעשה על סמך המרחק קוסיין בין ייצוג השאלה לייצוגי הצ'אנקים (כלומר אמבדינגס). כלומר בוחרים כמה צ'אנקים הקרובים ביותר לשאלה מבחינת מרחק זה.

גישה זו עלולה להיות בעייתית גם כן כי לא תמיד מרחק קוסיין בין הייצוגים משקף את רלוונטיות של צ'אנק לשאלה. המאמר שנסקור היום מציע בנוסף לצ'אנקים לתת ל-RAG את הזכרון המאחסן את הייצוגים של השאלות האחרונות(או/ו השכיחות) ובנוסף לכל שאלה מחזיק סוג של KV-cache עבור השאלה הזו (מניחים שיש לנו דאטהסט המכיל שאולות ותשובות וגם דאטהסט D). אז KV-cache הזה הייצוג של וקטורי Key and Value עבור שכבה מסוימת (לקראת הסוף המודל וזה אחד הייפרפרמטרים של השיטה). KV-cache יעזור לנו לבנות תשובה בצורה טובה יותר.

אז איך כל העסק הזה עובד? במהלך האימון אנו לוקחים שאלה ותשובה מהדאטהסט של שאלות ותשובות ובאמצעותו בונים את ה-KV cache של המודל כי אנחנו יודעים מה הצ'אנקים הרלוונטיים ביותר לכל שאלה. הרי לכל צ'אנק אנו שומרים את ה-KV שלו (מחושב כאשר הצאנק מוזן למודל יחד עם השאלה).

עכשיו אנו רוצים לאמן את הרשת לנצל את ה-KV caches האלו בצורה יעילה. בשביל כך באימון לכל שאלה לוקחים את צ'אנקים הכי קרובים אליה (מבחינת האמבדינג), לוקחים את ה- KV cache עבורים ומאמנים את השכבות האחרונות של המודל להוציא את התשובה הנכונה. כלומר לומדים איך לשלב את התוצאה (attention maps) מהשכבות התחתונות יחד עם ה-KV cache שצברנו מהזכרון (יש עוד איזה שכבה לינארית מאומנת בנוסף). עדכון הזכרון מתבצע בצורה די סטנדרטית (LRU ובנוסף השכיחות נלקחת בחשבון).

האינפרנס עובד באותה הצורה פחות או יותר. בגדול המאמר מציע שיטה לשדרוג RAG באמצעות ניצול המצב של KV-cache במהלך האימון. די נחמד מודה…

https://arxiv.org/abs/2408.16967