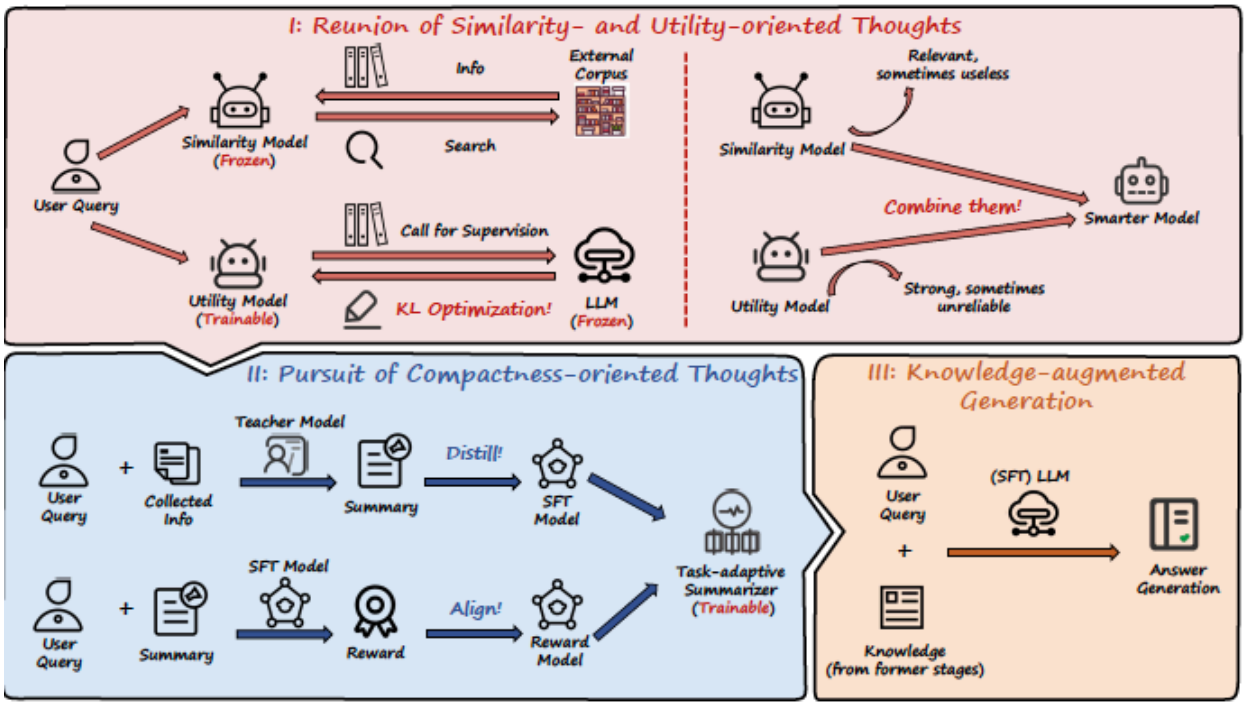
Similarity is Not All You Need: Endowing Retrieval-Augmented Generation with Multi–layered Thoughts



בזמן האחרון גישות המשלבות מודלי שפה עם בסיסי נתונים חיצוניים הפכו למאוד פופולריים. גישות אלו לרוב שייכות למשפחת Retrieval Augmented Generation או RAG בקצרה. בגדול בהינתן מודל שפה ומסמכים העשויים להכיל תשובה על שאלת משתמש, RAG קודם מחפש כמה מסמכים הרלוונטיים ביותר לשאלה ואז מזינה אותם יחד עם השאלה למודל שפה. המודל מרכיב את תשובתו על השאלה בהתבסס על המסמכים שהוזנו אליו.

אבל איך נבחר מסמכים הרלוונטים יותר לשאלה? בדרך כלל בוחרים אותם לפי הקרבה של האמבדינג (= ייצוג וקטורי) שלו לאמבדינג של השאלה. בדרך כלל המציאות טיפה יותר מורכבת ממה שתיארתי: למשל אם המסמכים ארוכים צריך לחלק אותם לצ'אנקים אז הבחירה היא לפי דמיון האמבדינגס של הצ'אנקים לזה של השאלה. כמובן שיש עוד גישות.

הדמיון בין אמבדינגס בד"כ מחושב לפי דמיון קוסיין (זווית בין הוקטורים). האם הבחירה הזו היא אופטימלית - זו השאלה שהמאמר שנסקור היום מנסה לענות עליה.

כדי להבין האם הבחירה אופטימלית צריך להגדיר מדד אופטימליות. הרי בסופו של דבר מטרתנו היא לתת תשובה נכונה לשאלת המשתמש. המאמר טוען שבחירת מסמכים רלוונטים לפי דמיון אמבדינגס אינו אופטימלי בהתאם המדד הזה. אז המחברים מציעים גישה לשכלול הבחירה של המסמכים הרלוונטים לשאלה.

האמת הם מציעים משהו די טבעי - בגדול המטרה שלהם היא לאפטם את הביצועים של RAG דרך "מקסום הסיכוי לקבלת תשובה טובה אחרי בחירת מסמכים רלוונטים על ידי RAG". המחברים מנסים להשיג את המטרה בכה שלבים:

שלב 1: אימון מודל utility. המטרה של מודל זה להעניק ציון ליכולת של מסמך נתון "לתת' תשובה טובה לשאלה כאשר הם (המסמך והשאלה) מוזנים למודל שפה יחד. אבל איך נדע לשערך את איכות התשובה? בשביל זה המחברים לקחו מודל שפה חזק (נגיד gpt4) שמטרתו היא לתת ציון לתשובה עבור מסמך ושאלה נתונים (ככל שהתשובה טובה ציון גבוה יותר). המאמר לא מסביר איך זה נעשה אבל אני מניח שעבור דאטהסט המכיל תשובות ניתן למדוד דמיון סמנטי בין תשובה אמיתית לתשובה מופקת על ידי llm (כלומר בין האמבדינגס), ניתן גם למדוד אותה על ידי הזנתם של המסמך, השאלה והתשובה ל-llm ומדידת נראות מירבית שלה (כלומר logits), בטח יש עוד שיטות. המחברים מאמנים utility model (שהוא מודל קל יחסית) להחזיר את אותה ההתפלגות של ציוני מסמכים (בהינתן שאלה) כמו המודל החזק. כלומר ממזערים KL divergence בין התפלגות ציונים של utility model לבין זו של מודל השפה (שהוא מוקפא - לא מאומן).

שלב 2: בחירת מסמכים עבור שאלה נתונה בוחרים רק מסמכים שיש להם ציון דמיון או ציון של utility model גבוה מספיק (בין k הגבוהים ביותר כל אחד).

שלב 3: אימון מודל תמצות מסמכים. המחברים טוענים שבד"כ המסמכים שנבחרים מכילים לא מעט מידע לא רלוונטי לשאלה שמקשה על מודל שפה לתת תשובה טובה וגם מעלה עליות (צריכים להכניס הרבה טוקנים ל-LLM). במטרה להתמודד עם הקושי הזה המחברים מציעים לאמן מודל שבהינתן שאלה מפיק מהמסמכים שנבחרו את המידע הרלוונטי לשאלה. זה נעשה ב 2 שלבים: בשלב הראשון עבור דאטהסט של שאלות והמסמכים הרלוונטיים מתשאלים מודל שפה חזק (gpt4) לתמצת את המסמכים האלו (עבור שאלה נתונה). על הדאטהסט הזה (שאלה, מסמכים ותמצית) עושים פיינטיון של מודל שפה לא כבד עם LoRa כמובן - כלומר עושים Supervised Fine-Tuning או SFT. בשלב השני עושים RLHF עם DPO כמו שמקובל היום 🙂. בשביל באמצעות מודל שפה(הם לא מפרטים יותר מדי כאן) בונים דאטהסט של תשובות נכונות ולא נכונות בהינתן שאלה ותמצית מסמכים. בניית פונקציית תגמול (reward) מתבצעת בדיוק כמו ב- DPO הסטנדרטי.

אחרי שסיימנו לאמן את מודל התמצות, ההיסק (אינפרנס) נעשה בצורה מאוד טבעית. לוקחים שאלה, מפיקים את המסמכים הרלוונטיים משלב 1, מתמצתים אותם עם המודל משלב 3 ואז מזינים אותם לעוד מודל שפה (המחברים לא מפרטים עליו אבל מציינים שניתן לכייל אותו על דאטהסט כלשהו של שאלות ותשובות). והמודל מספק לנו את התשובה...

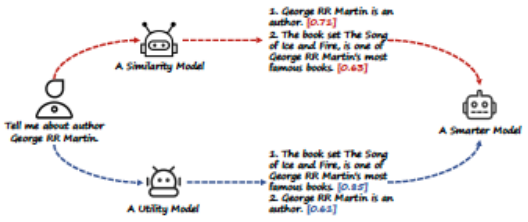


Figure 1: A toy example illustrating the difference between similarity and utility, where the score of similarity model is given by BGE[1]. Can we reunite the virtues of both worlds and come up with a better model?