**הקלות הבלתי נסבלת של המחיקה: מאמר כחול-לבן מהממציא של flow matching  
המאמר היומי של מייק: 24.09.25  
  
Corrector Sampling in Language Models**

מודלי שפה אוטורגרסיביים, המנועים המניעים את AI כרגע, בנויים על הנחת יסוד פשוטה, עוצמתית, אך פגומה מיסודה: עריצותו של חץ הזמן. הם כותבים טקסט טוקן אחר טוקן, משמאל לימין. כל מילה חדשה נבחרת על בסיס הרצף שקדם לה. מרגע שמילה מונחת, ההחלטה היא סופית, בלתי הפיכה. אין backspace.

זהו החטא הקדמון של יצירה אוטורגרסיבית. בחירה אחת, תת-אופטימלית במקצת, שם תואר לא במקום, פועל מסורבל, אינה רק טעות מקומית. זהו זרע מורעל. המודל חייב לאחר מכן להתנות את כל בחירותיו העתידיות על שגיאה זו, מה שמוביל למפל של אי-דיוקים מצטברים. הטקסט שנוצר עלול לסטות, לאבד קוהרנטיות, או להיכשל בלכידת הנתיב האופטימלי במרחב העצום של רצפים אפשריים. במשך שנים, הפתרון היה לבנות מודלים גדולים יותר עם נתונים טובים יותר, ולמעשה לנסות להפוך אותם לבעלי ראיית נולד עד כדי כך שלעולם לא יעשו טעות. אבל זה לא מתקן את החולשה המבנית הבסיסית.

המאמר כחול-לבן מציג אתגר אלגנטי מבחינה רעיונית ומבוסס מתמטית לפרדיגמה זו. הוא אינו מציע ארכיטקטורה חדשה או דאטהסט חדש. במקום זאת, הוא חושב מחדש על עצם פעולת הדגימה. המחברים מציגים שיטה בשם Resample-Previous-Tokens או RPT אשר, במהותה, מעניקה למודל backspace מקומי.

### **ההצצה הא-סיבתית**

הרעיון המרכזי פשוט עד כדי הטעיה. כאשר המודל מייצר טקסט, הוא לא רק מניח טוקן וממשיך הלאה. במקום זאת, הוא עוצר, מביט לאחור על חלון קטן של הטוקנים שנוצרו לאחרונה (נניח, השניים האחרונים), ושואל סוג חדש של שאלה. במקום לשאול רק, "בהינתן העבר, מהו הטוקן הבא?", הוא שואל גם, "בהינתן העבר *וגם* הטוקן שבא אחריו, מה היה צריך להיות הטוקן הזה?"

זוהי פריצת הדרך הרעיונית. המודל מאומן לא רק על חיזוי-הטוקן-הבא (NTP) הסטנדרטי, אלא גם על חיזוי-הטוקן-הקודם (PTP). התובנה התיאורטית המרכזית של המחברים היא ששגיאת החיזוי עבור PTP נמוכה מטבעה מזו של NTP. הידיעה מה יבוא בהמשך מספקת רמז קונטקסטואלי רב עוצמה המפחית את אי-הוודאות לגבי מה שקדם לו. זה כמו לנסות לנחש מילה במשפט; קבלת המילה שאחריה מהווה יתרון עצום.

במהלך הדגימה, RPT משתמש ביתרון זה כדי ליצור לולאת-מתקן זעירה. הוא מייצר טוקן חדש, ואז נכנס מיד לתהליך איטרטיבי שבו הוא חוזר לטוקן הקודם, ודוגם אותו מחדש על סמך ההקשר החדש שמספק יורשו. חילוף זה יכול לחזור על עצמו, ומאפשר לזוג הטוקנים להתייצב בתצורה יציבה יותר ובעלת סבירות גבוהה יותר, לפני שתהליך היצירה ממשיך הלאה.

### **המתמטיקה של התיקון**

בעוד שהמאמר עשיר בפורמליזם מתמטי שאני אישית מאוד מעריך, הטיעון המתמטי שבבסיסו ניתן להבנה אינטואיטיבית. תהליך הדגימה האיטרטיבי בתוך חלון ה-RPT יכול להיות ממוּדָל כשרשרת מרקוב. ה'מצבים' בשרשרת זו הם רצפי הטוקנים האפשריים בתוך החלון, והמודל עובר בין מצבים אלה על ידי דגימה מחדש של טוקנים. המטרה היא שהשרשרת הזו תתכנס במהירות להתפלגות הסטציונרית שלה, מצב של שיווי משקל המייצג בחירה גלובלית אופטימלית יותר של טוקנים מאשר מעבר קדימה יחיד יכול היה להשיג.

זה לא רק טריק הנדסי; זוהי גישה עקרונית להפחתת הצטברות שגיאות. על ידי מתן אפשרות לתיקונים מקומיים, המודל יכול לגזום נתיבים תת-אופטימליים בעץ היצירה לפני שיש להם הזדמנות להפיץ את שגיאותיהם הלאה. כמובן, מודל סטנדרטי אינו יודע מטבעו כיצד לבצע חשיבה א-סיבתית זו. כדי לאפשר זאת, יש ללמד את המודל במפורש להתמודד עם תמורות (permutations) מקומיות אלה. המחברים משיגים זאת באמצעות שלב קצר של פיין טיון, שבו מוצגים למודל סדרות עם טוקנים סמוכים שהוחלפו בכוונה. הדבר מאלץ אותו ללמוד את המיומנות החדשה של חיזוי זהותו של הטוקן בהינתן הקשר הכולל אלמנטים הן מעברו והן מעתידו המיידי. מכיוון שאימון מיוחד זה מעורבב עם חיזוי הטוקן הבא הסטנדרטי, המודל לומד את יכולת הדגימה החדשה והגמישה הזו מבלי להתפשר על נקודות החוזק המקוריות שלו.

במהלך שלב אימון נוסף וקצר זה, הם לוקחים משפטים רגילים מהדאטהסט אך בסבירות נמוכה, הם מחליפים בכוונה בין שני טוקנים סמוכים. כך, "החתול ישב על השטיח" עשוי להפוך ל"הישב חתול על השטיח" עבור דוגמת אימון ספציפית. כאשר המודל מקבל את הקלט שעבר תמורה ("הישב..."), מטרתו משתנה. הוא מתבקש כעת לחזות את הטוקן שהיה אמור להיות שם ברצף המקורי והנכון. הוא לומד שבהינתן ההקשר "ה" (עבר) ו"חתול" (טוקן "עתידי" בסדר המקורי), הטוקן שעליו לחזות עבור המיקום השני הוא "ישב".

למידה מתמורות על ידי חשיפה חוזרת ונשנית להחלפות מקומיות אלה וקבלת המשימה לחזות את הטוקן המקורי, המודל לומד את המיומנות החיונית החדשה של חיזוי-הטוקן-הקודם. הוא מפתח את היכולת להסיק מהו טוקן על סמך מה שהופיע לפניו ומה שהופיע מיד אחריו. "אימון ההחלפות" המיוחד הזה לא קורה בכל דוגמה. רוב הזמן במהלך פיין טיון המודל עדיין מאומן על NTP הסטנדרטי משמאל לימין. זה מבטיח שהוא לא ישכח את תפקידו העיקרי. התוצאה היא מודל ששומר על כל יכולותיו האוטורגרסיביות העוצמתיות, אך רכש מיומנות נוספת, כירורגית: היכולת לבחון מחדש ולתקן באופן מקומי את הפלט של עצמו על ידי מבט צעד אחד קדימה.

מה שהופך את העבודה הזו לכל כך משכנעת הוא ההתמקדות שלה בתהליך הדגימה עצמו, רכיב שבדרך כלל מתעלמים ממנו בנוף ה-LLM. היא מציעה שהמודלים שלנו עשויים כבר להחזיק בהבנה עמוקה יותר של השפה ממה ששיטות הדגימה הנוקשות שלהם, משמאל לימין, מאפשרות להם לבטא. RPT מספק מנגנון לשחרר את הפוטנציאל הזה, לא על ידי בניית מודל גדול יותר, אלא על ידי מתן אפשרות למודל הקיים לחשוב פעמיים. זהו צעד קטן לאחור שמבטיח קפיצת דרך משמעותית קדימה.

https://arxiv.org/abs/2506.06215