Reasoning Inconsistencies and How to Mitigate Them in Deep Learning

הרעיון המרכזי של המאמר

המאמר עוסק בשאלה - למה מודלי Deep Learning מצליחים מאוד במדדי ביצוע רגילים, אבל עדיין מייצרים reasoning לא עקבי, שגוי, לא יציב או לא ניתן לאימות.

הטענה המרכזית - הבעיה אינה רק שהמודל "טועה" לפעמים, אלא שיש לו דפוסים שיטתיים של חוסר עקביות.

כלומר, המודל יכול להיראות חזק ב - accuracy, אבל עדיין להיות:

- לא עקבי לוגית
- לא robust
- רגיש לשינויים קטנים
- מושפע מהטיות בדאטה
- לא faithful בתהליך ה - reasoning שלו

הקשר בין Deep Learning לבין Reasoning

במאמר, reasoning מוגדר כתהליך שבו מודל מקשר בין input לבין output דרך סדרה של transformations.
כלומר, במודלי Deep Learning, המודל לא "חושב" במובן אנושי, אלא מבצע רצף של שכבות וייצוגים פנימיים, שבסוף מובילים להחלטה.

הבעיה - ככל שהמודלים גדולים ומורכבים יותר, קשה יותר להבין האם הדרך שבה הם הגיעו לתשובה באמת נכונה.

מה הקשר בין DL לבין NLI / NLU

- **NLU, Natural Language Understanding -** תחום רחב בתוך תחום ה – NLP, העוסק בשאלה של "האם המודל באמת מבין שפה טבעית".
  הבנת השפה מתבטאת ב :
  - הבנת המשמעות
  - הבנת הקשר בין המשפטים
  - הבנה לוגית

  משמע, תחום ה – NLU הוא תחום שמטרתו לבחון את יכולת ה - Reasoning של המודלים.
  - **Reasoning –** תהליך החשיבה או ההסקה שבו המודל משתמש כדי להגיע ממידע נתון (input), למסקנה או לתשובה.
- **NLI, Natural Language Inference -** זו משימה שבה המודל מקבל 2 משפטים, Premise ו – Hypothesis, והוא צריך לקבוע מה טיב הקשר ביניהם (contradiction / entailment / neutral).

ובהקשר של המאמר, NLI מוצגת כאחת המשימות המרכזיות לבדיקת reasoning ב - NLU.

NLI הוא תחום חשוב היות והוא לא רק בודק האם המודל מזהה מילים, **אלא גם** האם המודל מבין קשר לוגי בין שני משפטים.

ולכן - אם מודל NLI משנה prediction בגלל paraphrase קטן (שינוי קטן בהיפותזה), זו אינדיקציה לכך שה - reasoning שלו לא יציב (משמע, בעיית ה – reasoning inconsistencies).

שלושת מקורות הבעיה לפי המאמר

המאמר מחלק reasoning inconsistencies לשלושה מקורות מרכזיים:

1. Internal Processes - חוסר עקביות שנובע מהתהליכים הפנימיים של המודל.
כלומר - המודל למד shortcut או heuristic פנימי, ולכן הוא משנה החלטה בגלל שינוי שטחי.
   a. הערה שלי (לידור) – ביחס למאמרים הקודמים שנסקרו, וביחס למאמר ה Anchor שקיבלנו (Semantic Sensitivities and Inconsistent Predictions), נשמע שזו הסיבה העיקרית.
   הרוב המוחלט של הסקירות הקודמות מראות כי הוא מראה שמודלי NLI רגישים לשינויי ניסוח קטנים, גם כאשר המשמעות נשמרת.

2. **Data Imbalances** - חוסר עקביות שנובע מהדאטה שעליו המודל התאמן.
כלומר - אם הדאטה לא מאוזן לפי labels, topics או semantic patterns, המודל עלול ללמוד הטיות במקום reasoning כללי.
   a. **לדוגמה** - אם topic מסוים מופיע כמעט תמיד עם label מסוים, המודל עלול ללמוד את ה - topic במקום להבין את הטענה.
   b. הערה שלי (לידור) – נדמה שכל הדאטה-סטים שעבדו איתם בסקירות הקודמות, לא ציינו חוסר עקביות. כן היה דיבור שם על in / out domain, אבל לא היה דיבור על imbalance בתוך כל domain. כן אני אגיד שאם היה imbalance בכל דומיין, אז היינו מצפים לתוצאות Accuracy גרועות, וזה לא מה שהמאמרים מציגים.

3. **Task Complexity** - חוסר עקביות שנובע מכך שהמשימה עצמה מורכבת מדי.
כלומר, המודל צריך לבצע reasoning מרובה שלבים, לשמור מידע ביניים, ולחבר כמה מסקנות יחד.
במקרים כאלה, המודל יכול להגיע לתשובה נכונה, אבל עם reasoning path לא נכון או לא ניתן לאימות.
   a. **הערה שלי (לידור) –** פה נכנס הרעיון של רומי, לשלב LLM לפני ההכנסה למודל, כדי לפרק את המשפט, לשלוף ממנו נתונים, ואז מודלי ה – NLI יקבלו את התוצר של ה – LLM.

סוגי מודלים / שיטות / ארכיטקטורות מופיעים במאמר

המאמר מתמקד בכמה משפחות של מודלים ושיטות :

- **Transformer-based Language Models**
מודלים כמו RoBERTa, BART, DeBERTa ו - DistilBART.
  - הערה (לידור) - אלו המודלים המרכזיים בפרק של NLI וה - Anchor Paper.

- **LLMs**
מודלים גדולים המשמשים ל - QA, reasoning ו - generation.
הבעיה המרכזית בהם היא שהם יכולים להפיק reasoning כתוב שנראה משכנע, אבל לא בהכרח faithful.

- **Vision Models**
מודלים לעיבוד תמונה, בעיקר סביב adversarial robustness ו - pretrained backbones.

- **Knowledge Graph Models**
מודלים שמבצעים reasoning על גרפים של ישויות וקשרים.
כאן מופיע עיסוק ב - Complex Query Answering.

- **Neuro-symbolic / Logic-Aided Methods**
שיטות שמשלבות LLMs עם פורמליזציה לוגית, כמו Prolog-like reasoning.

פתרונות שהמאמר מציע

בניגוד להרבה מאמרים שרק מודדים את הבעיה, המאמר מציג גם ניסיונות mitigation.

הפתרונות המרכזיים:

- **Semantic Sensitivity Framework** - מדידה של חוסר עקביות ב - NLI באמצעות paraphrases ששומרים על משמעות.

- **TESTED** - שיטה ל - Topic-Guided Sampling.
הרעיון הוא לבחור דוגמאות אימון כך שהדאטה יהיה מאוזן יותר לפי topics, ולא רק לפי labels.

- **SynDARin** - שיטה ליצירת datasets סינתטיים עבור שפות low-resource.
הרעיון הוא לאפשר בדיקת reasoning גם בשפות שאין בהן הרבה דאטה.

- **CQDA** - שיטה לשיפור Complex Query Answering על Knowledge Graphs.
הרעיון הוא לכייל intermediate scores כך שהשלבים הלוגיים יעבדו טוב יותר יחד.

- **FLARE** - Faithful Logic-Aided Reasoning and Exploration.
הרעיון הוא לגרום ל - LLM לפרק בעיה, לנסח אותה בצורה לוגית, ולבצע simulated search שניתן לבדוק אותו. זה כנראה הפתרון הכי חשוב עבורנו, כי הוא מתמודד עם הבעיה של reasoning לא faithful.

הרחבה על - FLARE

הבעיה - LLMs יכולים לכתוב Chain-of-Thought יפה, אבל לא ברור אם ה - chain הזה באמת גרם לתשובה.
כלומר - ההסבר יכול להיות בדיעבד, ולא בהכרח reasoning אמיתי.

FLARE מנסה להפוך את תהליך ה - reasoning של המודל ליותר:

- **Structured** - reasoning מסודר לפי שלבים ברורים, ולא תשובה שנזרקת בבת אחת.
- **Interpretable** - כזה שבני אדם יכולים להבין ולעקוב אחריו.
- **Verifiable** - reasoning שאפשר לבדוק האם הוא נכון לוגית.
- **Faithful** - שהשלבים שהמודל מציג באמת שימשו אותו להגיע לתשובה, ולא explanation שנכתב רק בדיעבד.

FLARE עושה זאת באמצעות:

- **Plan** - פירוק הבעיה לשלבי reasoning מסודרים.

- **Code / Logical Formalization** - המרת חלקים מהבעיה לייצוג לוגי או פורמלי, שקל יותר לבדוק ולאמת.

- **Simulated Search** - ביצוע חיפוש או exploration של כמה מסלולי reasoning אפשריים, במקום להסתמך על תשובה אחת מיידית.

- בדיקת התאמה בין ה - reasoning trace לבין הפתרון

- בדיקה האם הדרך שהמודל הציג באמת מובילה לוגית לתשובה הסופית.

Future Work שמופיע במאמר

ה - Future Work המרכזי מדבר על הרחבת reasoning בזמן inference.

כלומר, לא רק לאמן מודל טוב יותר, **אלא** לתת לו יכולות חישוב ובדיקה בזמן שהוא פותר את הבעיה.

הכיוונים המרכזיים שמוצעים:

- **Test-Time Compute -** להשתמש ביותר חישוב בזמן פתרון הבעיה, כדי לאפשר למודל לבדוק, לשפר ולתקן את ה - reasoning שלו.
- **Self-Refinement -** המודל מייצר תשובה, בודק אותה, ומנסה לשפר אותה באופן איטרטיבי.
- **Reward Model based on FLARE -** לאמן מודל שמעריך reasoning paths.
  הרעיון הוא מסלולי חיפוש לוגיים נכונים ישמשו כדוגמאות חיוביות, ומסלולים שגויים של LLM ישמשו כדוגמאות שליליות.
- **Differentiable Oracle -** לבנות מנגנון שיוכל להעריך האם reasoning path נכון ושלם, ואז להשתמש בו כדי לשפר מודלים אחרים.
- **Iterative Multi-Hop Reasoning -** פירוק בעיה לכמה שלבי reasoning, חיפוש תשובות ביניים, וסינון כיוונים לא סבירים.

למה זה חשוב למחקר שלנו

עד עכשיו המחקרים שסקרנו התמקדו בעיקר ב"האם מודלי NLI עקביים תחת שינויי ניסוח".

המאמר הזה מרחיב את התמונה - הוא אומר שהבעיה של NLI היא רק מקרה אחד בתוך בעיה רחבה יותר של Deep Learning :

- מודלים מצליחים מבחוץ, אבל תהליך ה - reasoning שלהם לא תמיד עקבי, אמין או ניתן לבדיקה.