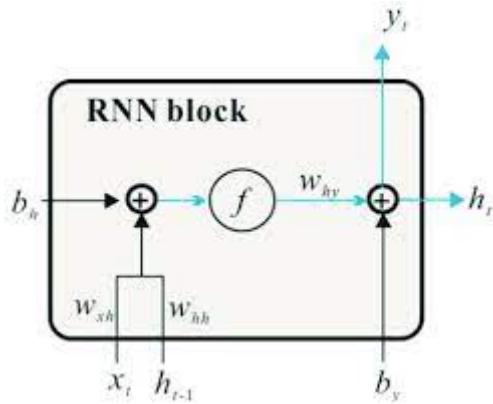


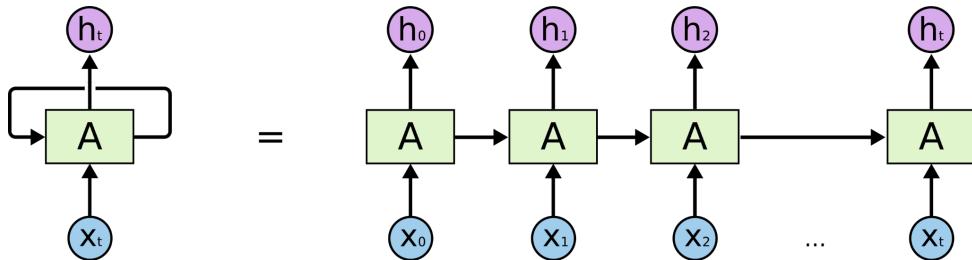
לפנינו שנדון בפתרונות שהוצעו לבעה, נפתח בהצגה שלה. כאשר אנו מדברים על בעיה, עלינו להתייחס למורכבות (complexity) שלה. מורכבות של משימה הנשענת על ניתוח נתונים יכולה להתבטא בכמות המימדים, גודל הקלט, רעש, דינמיות חריגות (outliers), ודפוסים מורכבים החובים בדעתה. בהקשר זה, שפה טבעיות היא בעיה עם מורכבות גבוהה, שכן היא מכילה תלות ארוכה וקצרת טוח בין מרכיבי הקלט השונים, סלנג, כפל שימושות (כדוגמת סרקיום), וסמנטיקה חבויה ונולית. שפה מקיימת גם כליל תחבורה ודקડוק, כאשר שימושות של מילה מסוימת תלויות בהקשר בו היא נאמרת, ועל כל אלו שפה אינה שומרת על רציפות באופן מוחלט. אלגוריתמים בעלי חוקיות מוגדרת מראש (rule-based) כדוגמת Cyc שנייה לפורמל פתרון לנитוח שפה באמצעות סט חוקים פנימיים, נכשלו בעקבות כך.

נדנים創נים את מורכבות ניתוח שפה, באמצעות בעיית התרגומים, ואת הקשיים שהוא מציפה. נניח ואנו מעוניינים לתרגם את המשפט הבא מאנגלית לעברית: "The dog sleeps" - "הכלב ישן". ניתן לתרגם כל מילה במשפט ללא תלות במילים האחרות. הכלל היחיד שעלינו להתחשב בו הינו שהמילה "the" תוסיף תחילית "ה" למילה שבאה אחריה. ولكن, נוכל לשמר מילון המכיל לכל מילה באנגלית את התרגום שלה בתוספת לחוקים שהצענו ולקבל תרגום תקין. אולם, בהינתן המשפט "We're all mad here" (עליסה בארץ הפלאות) תוצאת התרגום שבוצע בהתאם לחוקים אליו תראה "colsno את כל מטורף כאן". שורש הבעיה הוא תרגום המילה הנוכחית ללא התחשבות בהקשר(context), לעומת במהלך התרגומים אנו חיברים להתחשב בתלותות בין המילים שאנו מתרגמים לבין המילים האחרות במשפט. ניתן להציג פתרון המשך למלון שהצענו, ולשמור בנסוף על המילים והתרגומים שלהם, גם צמדי מילים, אך זה פתרון נקודתי לבעה מבנית, וכי מהר נצטרך להרחיב את המילון לביוטים בעלי 3, 4 ו יותר מילים. ולבסוף, נתכנס לכישלון המשותף לכל האלגוריתמים שניסו לפתר את הבעיה בצורה (rule-based), והוא חוסר יכולת למדל את חוקיות הבעיה בצורה ייעלה.

הארQUITטורה הריאונית שניסתה למדל את התלות בין מילים בקלט מתוארכת לסוף שנות השמונים של המאה הקודמת [2], ונקראת RNN. הרעיון המרכזי מאחורי הוא ניתוח הקלט הנוכחי, בתוספת למידע שהתקבל מהWORDS האחרות במשפט. הרשת מקבלת מילה במשפט ומיצרת שני פלטים, הראשון הוא פלט המתאים למילה ביחס למשמעותה (תרגום, תקצר טקסט, יצירט טקסט וכו'), והשני הוא ההקשר שנוצר עד עתה ומיצג תמצאות של כל המידע שהרשות קיבלה עד אז (ביצוג lossy). פתרון פשוט לבעה זו הינו שרשור של יחידות חישוב כ אלה, כך שכל יחידת עיבוד (שנקרא לה בלוק מטמוני נוחות החל מכאן) מקבלת קלט בזמן t (כלומר את הקלט במקומות -t במשפט/قطع טקסט) ואת ההקשר מזמן t-1 ומיצרת את הפלט המתאים לו. אולם, משפטיים שונים מכילים מספר מילים שונה, ולכן, לא ניתן להחזיק מספר בלוקי מוגבל של בלוקים כך שכל בלוק יתאים לקלט מסוימת במשפט. בנוסף בארQUITטורה שכמו הблוקים הראשונים יؤمنו הרבה יותר מהблוקים האחרונים, דבר הנובע מכך שבמשפטים, קיצרים ממספר הблוקים המרבי, וחופדו באפסים, והблוקים האחרונים לא ישתתפו באופן מפורש בתהליכי הלמידה. ולכן, הפתרון הינו הפעלה איטרטיבית של הרשת. כלומר שימוש בבלוק יחיד וכתוכאה מכך בסיס משקלות מסווג, כך שהרשת מקבלת בכל איטרציה קלט חדש ואת ההקשר שהתקבל במצבה הנוכחי. הייצוג הטורי מהוווה פרסה(unfolding) של הרשת (ראו איור 2), ביצוג הקומפקטי: הרשת מקבלת קלט X ואת המצב הפנימי מהאיטרציה הקודמת שמתואר כ- H_{i-1} , ומוציאה פלט Z, בעוד שה המצב הפנימי H מוחזר בחזרה לרשת באיטרציה הבאה.



איור 1: מבנה רכיב RNN



איור 2: RNN ביצוג טורי (מיימי) וביצוג קומפקטי (unfolded)

אז כיצד משתמשים ברשת כזו לניתוח ניתוח שפה? טרם השימוש מקודדים את המילים/חלקי מילים (word/subword embedding). פועלות הקידוד היא המרת מילה בשפה טבעית, לווקטור בעל K ממדים, כאשר הייצוג הווקטורי מקובץ מילים בעלות משמעויות דומות תחת ערכיהם קרובים, ומרחיק מילים בעלות משמעויות שונות אחת מהשנייה. [בקיים](#) הנויל נתן לראות שיטות שונות לבצע זאת, כאשר הנפוצה ביותר היא שימוש ברשת נירונים (לדוגמה word2vec). הבועה באלגוריתמים אלו, הינה שקידוד מילים בעלות כל משמעות יהיה זהה, דבר שמקשה על הלמידה. בהקשר זה, אחד החידושים שאריכטקטורת הטרנספורמרים הצינה היה שילוב שיכון (embedding) המילים כחלק מהאריכטקטורה, דבר שאפשר למידה של שיכון המילה בגין מילים קרובות למילה הנוכחית, אלא גם מילים רחוקות ממנה (כלומר שיכון מילים במרחב תלו依 הקשר או contextualized embeddings).

על פניו, נראה שהרשות מספקת פתרון לבעה, היא מסוגלת לאנור מידע ולמודד את התלות בין חתימות הזמן השונות. איז מדוע זה לא עובד? ישנן מספר סיבות האחריות לכך. הראושונה נעוצה במבנה הטורי של הרשת, שהוביל כמעט תמיד לדעיכת הנגדיאנט לאפס (vanishing gradient). כפי שהוזכר בהתחלה, במקרים מסוימים ניתוח שפה הפלט עבר קלט מסוים תלוי בסדרת קלטים שכברלו לו, ולכן, בעת אימון הרשת המשקولات מעודכנות רק לאחר העברת הסדרה כולה. בנסיבות איטרטיביות, יש לחשב את הנגדיאנט של המשקولات גם כן בסדר איטרטיבי אולם הפעם הוא הפוך. כדי לחשב את השפעת המשקولات על השינוי בשנייה, علينا לכפול בטור את הנגזרות החלקיות של כל מצב פנימי ? בגין מצב הפנימי 1-. מכיוון שהשינוי מחושב עבור אותו סט משקولات, הנגדיאנט יכול לדעוך או להתבדר לאורכו האימון. ניתן להמחיש זאת באמצעות הערכים העצמיים של מטריצת המשקولات; במקרה והם קטנים מ-1, הנגדיאנט ידוע מכיוון שמטריצת המשקولات מקטינה אותו בכל פעם שמתבצע עדכון, ובמקרה והערכים העצמיים גדולים מ-1 הנגדיאנט יתבדר מכיוון שמטריצת המשקولات מגדילה אותו בכל איטרציה ([דיעיכת הנגדיאנט ופתרונות RNN](#)). הבעה השנייה נובעת מהיחס בין גודל המצביעים הפנימיים והיכולת שלהם לדוחס מידע, לבין גודל הסדרה שהרשות נדרשת לנתח. המצב הפנימי "דוחס" את כל המידע עד לנקודת הזמן הנוכחית, ולכן, כאשר

הסדרה עוברת סדר מסויים בקצבה ומתחילה להווצר תליות ארוכות טווח בין מרכיביה, המצב הפנימי הופך להיות צואר-בקבוק בראשת והיא מתחליה "לשוכח" פרטימ שרחוקים מהקלט הנוכחי שלה. הבעיה האחרונה צצה במהלך השימוש בראשת לאחר האימון (inference). הרשת אינה מנצלת את מלאה הפוטנציאל החישובי שלה (ושל המכונה שהיא רצה עליה) כתוצאה מאופי קבלת הקלטים באופן טורי, דבר הגורז זמן הסקה איטי (כזכור latency נבוה) של המצביעים הפנימיים במגוון שימושות NLP. בעיה מהותית זו נפתרת באמצעות הטרנספורמים שנזכר לעילו בהמשך.

נקודות מפתח לסיקום הפוסט:

- אלגוריתמים מבוססי חוקים (rule-based) נכשלו במשימות של ניתוח שפה טבעית עקב הקושי למדל בצורה אפקטיבית את הקשרים שפה טבעית מכילה. וכך, הפתרון הוא רשת שלומדת את הקשרים הללו.
- הרשת יכולה ייחודה עיבוד ייחודה המקבלת את המצב הפנימי הקודם שלה כמצב הפנימי החדש בתוספת קלט חדש, ולומדת יציג מותמצת של הקלט שראתה עד לנקודת הזמן הנוכחיית.
- הרשת סובלת משני בעיות עיקריות: דעיכת גרדיאנט לאורך האימון, ואי יכולת ללמידה תלויות ארוכות טווח.
- במהלך השימוש בראשת(במהלך ההסקה), בגלל אופי המידע הטורי, הרשת לא מנצלת את מלאה המשאים העומדים לרשותה, וכתוצאה לכך זמן עדכון המצב הפנימי איטי.