#### <u>מבואות</u>

מידול שפה- בהנתן רצף מילים מה היא המילה הבאה  $p(w_1|START)p(w_2|w_1) \dots p(w_k|w_1, \dots w_{k-1})$ ...  $p(w_N|w_1,...w_{N-1})$ 

קורפרנס-ביל אמר ששרה תגיע בקרוב, והיא באמת הגיע. היא ו- שרה מתייחסות לאותו אדם. אנאפורה -קודם אומרים את הביטוי ואז את האזכור קאטפורה-קודם האזכור ואז הביטוי קאטפורה-קודם האזכור ואז הביטוי

מרחק עריכה- מטריקה להשוואה בין מחזורות, חץ למעלה הוספת תו, חץ ימינה למחוק תו, חץ אלכסון הצצנו במה קיבלנו (משאירים או מחליפים)- לבחור את המינמום מבין השלושה

[^].(w או W) ביטוי רגולרים- [Ww] או על מה שבפנים שלילה על מָה שבתוך הסוגריים, או , ? תו קודם אופציונלי, \* אפס או יותר חזרות של התו קודם, + אחד או יותר חזרות של התו הקודם, - כל תו אפשרי, ^ מה שבא אחרי צריך להיות בתחילת המשפט, \$ אותו דבר רק על סוף משפט, ./ הופך את הסימן ( במקרה הזה . לחו רגיל

tr 'what to change' 'change to what' – UNIX פקודות - <file.txt> - לדוגמה: 'a-z' - a-z' - מחזירים את - <file.txt המילים השונות בין המסמכים, grep מחזיר את המילים התואמות לביטוי הרגולרי

**טוקן (תמנית) –** לעבור אחת אחת על המילים ולספור יייכן **Type (תבנית)-** לספור את הכמות הטוקנים השונים שיש

בטקסט (זה אוצר המילים בקורפוס) צורת יסוד- lemma- צורת היסוד של מילה – "צורת" -

be- " are,is" , הלכתי" – הלך, כדי לדעת מה הנושא של טקסט מסויים יעניין אותנו הלמה ולא הטוקנס

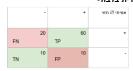
איך נדע את צורת היסוד של מילה – ( stem) איך נדע את בשפה שאנחנו לא יודעים? נסתכל ברמה השטחית על stori גבעול story צורת יסוד stories - - הטקסט

. TTR – היחס תבניות∖תמניות התפלגות מילים בטקסט גדול היא זיפפיאנית: עובר

 $p(w_n) \propto \frac{1}{n^s}$ המילה ההסתברות שליחותה ההסתברות שליחותה

#### סיווג מסמכים

c שהוא רצף תווים ואוסף תגים dפלט- תג רצוי מטרית בלבול-



 $\frac{TP+TN}{TP+TN+FP+FN}$  - Accuracy  $\frac{TP}{TP+FP}$  can nail't else $\frac{TP}{TP+FP}$  - בייודל תופס

כמה המודל תופס  $\frac{TP}{TP+FN}$  כמה מודל תופס תמרכב.  $\beta$  ממוצע הרמוני בין  $\beta$  ל  $\beta$  פרמטר  $\beta$  , הערך  $\beta$  את מגדילים את פרז לחזק את  $\beta=1$  המוזן

לחזק את R מגדילים א
$$F_{\beta} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

סיווג להרבה תגים(מאקרו)- חישוב P,R,Fעבור כל אחד מהתגים ואז נמצע את F, מטיבה עם הקלאסים הקטנים (התגים עם מספר איברים קטן) מיקרו – נחשב R,P עבור הכל ביחד ונחשב F אחד,

הקלאסים הגדולים מקבלים יותר משקל

פירת מילים. פיצרים אפשריים במערכת סיווג: ספירת מילים.

רשימות מילים המוגדרות כבעלות תכונה- ניתן לספור הופעות, מספר types, מימני פיסוק, מבנים תחביריים מידול שפה)- מחשיבים רק את ה n-1 המילים (מידול שפה) n-gram שקדמו למילה ה*ח* בשביל לחזות אותה

 $\hat{P}(w_k|w_{k-n+1},\ldots,w_{k-1}) = \frac{\text{COUNT}(\langle w_{k-n+1},\ldots,w_{k-1},w_k\rangle)}{\sum_w \text{COUNT}(\langle w_{k-n+1},\ldots,w_{k-1},w_k\rangle)}$ קסא, מילים  $n_d$  עם מסמך – Naïve base

 $c = \{c_1, c_2, \dots c_k\}$  תגים פלט- פונקציית מסווג

 $P(c|d) = \frac{p(d|c)p(c)}{p(c)}$  - חוק בייס p(d)

 $argmax(\sum_{w \in d} log(P(w|c))) + logP(c)$  $\hat{P}(c_i) = \frac{\text{no. of docs in } c_i}{\text{of docs in } c_i}$ 

$$P(c_i) = \frac{1}{\text{no. of docs}}$$

$$\hat{P}(w_i|c_j) = \frac{\text{count}(w,c_j)}{\sum_{w \in V \text{ count}(w,c_j)}}$$

החלקה לפלוס 1 –  $count(w,c_j)+1$ 

 $\hat{P}(w_i | c_j) = \frac{\sum_{w \in V count(w, c_j) + |v|}$ 

במונה- כמה פעמים המילה הספציפית מופיעה בקלאס במכנה- מספר המילים בכלל בקלאס (עם חזרה) ועוד גודל אוצר המילים

אם מילה לא מופעיה באחד הקלאסים – זו בעיה, זה יאפס את הargmax , ההחלקה פותרת את זה

# מערכת סיווג

מרכיבים – 1.ייצוג לקלט פיצ'רים  $\hat{y}$  פונקציית הסיווג מחזירה. . פונקצית הפסד

4. אלגוריתם למידה (SGD)

 $\sigma(\theta \cdot f + b) = \hat{y}$ רגרסיה לוגיסטית- פונקציית סיווג  $\sigma(z) = \frac{1}{1 + \exp(-z)}$  -סיגמויד

 $L_{CE} = -\log(P(y|x)) =$ פונקציית  $-[y * \log(\sigma(\theta f + b)) +$ 

 $\frac{\partial L_{CE}}{\partial \hat{x}} = (\hat{y} - y)f_j$ גזרת פונקציית ההפסד- $\frac{\partial L_{CE}}{\partial x} = (\hat{y} - y)$ 

 $(1-y)\log(1-\sigma(\theta f+b))$ 



גולריזציה: לבלום במפורש את ערכי הפרמטרים וגם להתמודדות עם

 $L = L_{\it CE} + L_{\it reg}$ 

מענישה מודלים  $L_{REG}^{r}=\left|\left|\theta\right|\right|_{2}^{2}$  : $L_{REG}^{r}$ שהתרחקו מהראשית.

מענישה מודלים  $L_{REG} = ||\theta||_{1}$ : L1

שיותר מידי פרמטרים בהם אינם 0. מודל overfitting – התאים את עצמו יותר מידי למדגם

גרסיה מרובת קלאסים- נחשב את הציון לכל קלאס

בסוף ננרמל בסכום כל הציונים , נחזיק גם וקטור משקולות עבור כל קלא*ָ*ס  $softmax(z_i) = \frac{exp(z_i)}{\sum_{j=1}^k exp(z_j)} 1 \le i \le k$ 

יולנוי קלאם הפחד-

:overfitting

$$L_{CE}(y, y)$$

$$= -log \frac{\exp(w_k * x + b) = (\hat{y}_k)}{\sum_{j=1}^{K} \exp(w_j * x + b_j)}$$
מולטי קלאס עדכון

$$\frac{\partial L_{CE}}{\partial w_{k,i}} =$$

$$= -\left(\mathbb{I}\{y=k\} - \frac{\exp(w_k \cdot x + b_k)}{\sum_{j=1}^{K} \exp(w_j \cdot x + b_j)}\right) x_i$$

כל הפרמטרים מתעדכנים שכן softmax אף פעם לא

פרספטרוו- ננסה לייצר קו מבחיו ביו דוגמאות חיוביות אליליות, אם צדקנו אין עדכון אחרת נעדכן לפי הקלאס זנכון

# <u>תיוג רצפים</u>

מילים m בעל d מסמך פלט- m תגים

בעיות- תיוג חלקי דיבר, תיוג תכונות מורפותחביריות,

תכונות מורפותחביריות- מין, גיל, זמן, בניין, גוף,

voice (active/passive) חד/רבים, חלהי דיבר- open calss – שם תואר adi שם תואר . שם פרטי , verb אין, שם עצם noun פועל adv

, קריאות (היי, יו) (היי, יו) (מרי, יו) (can) aux מילות יחס adp (adp ) מילות יחס - **Closed calss** מילים שמחברות בין חלקי משפט שווים cconj(וגם) מילות חיבור משעבדות -כלומר מחברות ביו משפט (the) det , אשר)sconj חות חשוב לאחד יותר חשוב sconj pron כינויי גוף , (up in – wake up) particles ,num

(to,with) prep אר X סימבולים o **– Other clas**s), כל השאר Other class - PTE הקורפוס המסיבי הראשון הגדול באנגלית - PTE

יטריקות- תיוג רצפים: דיוק ברמת המילה

רון) על התגים (מאקרו ומיקרו) *F1* 

יהוי יישויות מטריקות: . טעינו ביישות – טעות בהכל

חישוב £1 לכל היישויות או לאחת

סכימת B-BIO תחילת ישות, I אמצע, O לא חלק שרשרת מרקוב- הנחת היסוד- כל מצב תלוי רק במצב

 $P(q_i=a|q_1...q_{i-1}) = P(q_i=a|q_{i-1})$ 

(π) צטרר גם את ההסתברות להיות במצב הראשוו מודל מרקוב נסתר HMM – מודל גנרטיבי, רואים רק את המילה (הפלט) ואנחנו רוצים להבין איזה מצב (חלק ריבר) יצר אותה. המצבים נסתרים הנחת יסוד- פלט תלוי רק במצב שלו

ההסתברות של רצף התגים בהינתן רצף המילים:

$$\widehat{t_{1:n}} = \underset{argmax}{argmax} P(t_1 \dots t_n | w1 \dots w_n) pprox \prod_{i=1}^n P(w_i | t_i) P(t_i | t_{i-1})$$
 אלגוריתם ויטרבי- תכנון דינמי  
מטרה – מציאת המסלול הכי סביר בין המצבים שייצרו

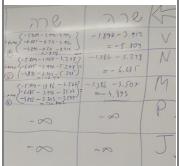
את התצפיות (המילים במסמך)

 $v_t(j) = \max_{i=1}^n v_{t-1}(i) \cdot a_{ij} \cdot b_j(o_t)$ t-1 געבור על כל המצבים האפשריים עבור התגt-1. 1

i הוא (t-1) הוא החשב את ההסתברות שהתג הקודם (t-1) הוא 2j נחשב את ההסתברות שמתג i עברנו לתג 3  $o_t$  נחשב את ההסתברות שמתג i יצא הפלט .4

נתעלם ממנו.  $v_{t-1}(i)$  ו"  $a_{i,1} = \pi(i)$  נתעלם ממנו.  $b_1(o_t)$ Start+emission - אלגוריתם - 1. עמודה ראשונה עמודה שניה והלאה- עושים בכל משבצת בטבלה מקסימום על כל אופציות המעבר מהעמודה הקודמת 

+ transision



Emissions (log)		שרה	שרה		שמח ש	
V		-3.912	-3.912 -6.21		-6.908	
N		-5.298	-4	.605		
M		-3.507	3.507 -3.912		-6.908	
P			-11.513			
J					-3.219	
Transitions (log)	-> V	-> N	→ M	.> P	» J	
V->	-2.9	96 -1.050	-1.386	-1.609	-1.897	
N->	-0.5	11 -2.996	-2.996	-2.303	-1.609	

-CRF מודל דיקרמינטבי. כל תג תלוי בתג הקודם במיקום עצמו ובכל הקלט (לצורך פיצ'רים)  $y_i$ , X, i

2.996 0.693

תבניות פיצ'רים מקובלות -

 $< y_{i'} x_{i} >, < y_{i'} y_{i-1} >, < y_{i'} x_{i'} x_{i-1} >$ 

רשת ניורונים רשת בהיזן קידמי FFN: המון שכבות של נוירונים פשוטים בתוספת פונקציית אקטיבציה ביניהם. MLP אותו דבר רק שהנוירון הפשוט הוא פרספטרון.

אקטיבציות אפשריות:

-1.966 -1.897

-2.99

1. RELU בחירה טובה, גזירה קלה

2. סיגמויד, גם אפשרי. (0-1) tanh.3 טנגנס היפרבולי. (1- ל1) אימוו הרשתות: מציאת הנגזרות

החלקיות לפי הLoss ומפעפעים אחורה back propagation והשם של זה הוא .(כלל השרשרת).

מגבלות הFFN: התייחסות לכל הקלט בבת אחת. לא מתחשב בסדר קלט.

מוגבל לחיזוי תג אחד. לא עובר מידע בין נוירונים באותה FFNב\*

השכבה, רק בין שכבות עוקבות, ואילו בRNN כן. LSTM- גרסה נוספת ומשופרת לRNN.

מכיל מרכיבים שנועדו לשמור על מצבים

Bidirectional RNN- מבדיל ביו הכיוונים. לכל כיוון הפרמטרים שלו והחיבור נעשה רק ברמה הבאה.

רגולריזציה – dropout – בכל שלב נאפס חלקים אקראיים בכל שכבה, אבל רק בשלב האימון.

# מידול מבנים ותלויות (תחביר)

 $N, \Sigma, R, S$ - CFG דקדוק חסר הקשר חומסקי CNF כל כלל כולל מימין שני נונטרמינלים או מטריקת פרסבל- לוקחים את כל המרכיבים בעץ ה"נכוו"

בודקים כמה המערכת זיהתה נכון (recall) לוקחים את (precision) מרכיבי העץ החזוי ובודקים כמה נכונים F1 מדווחים

מטריקת חציית סוגריים – יחס המרכיבים הנכונים בלי הפרעה במבנה.סוגר ] בלתי סגור בתוך[] אלגוריתם CKY – תכנות דינמי



#### つつか S NP VP D N NP chef D



סוגי קשתות בעץ תלויות-קשתות בין נשוא (פרדיקט) לארגומנט ליבה:

nshdi נושא שמני נושא פסוקי ללמוד הרבה לדגומה לדגומה משפט "ללמוד הרבה csubdj נושא זה חשוב" – ללמוד הרבה זהו הנושא הפסוקי

מושא ישיר dobj her – he gave her the book מושא עקיף iobj מושא iobjהוא ה

פסוקית משלימה xcomp , ccomp הוא אמר לה xcomp – לסדר את החדר" – לסדר את החדר

. המילה אתמול היא לא מרכיב ליבה הכרחי של המילה

הלר. כלומר גם בלעדיה המפשט תקיו פסוקית אופן *advcl* כמו המילים במהירות, ביעילות aux,cop פועל עזר

קשתות בתוך צירופים שמניים: the מו determiner מו סמיכות *nmod* כמו "כיתת לימוד'

case סחי מילת ניתוח תלויות במעברים- החלטות מקומיות, 3 מבנים: באפר – מחזיק את כל המילים שטרם

נויפלנו מחסנית- מועמדים לקבלת קשת

.3 הקשתות שהוספנו כבר לעץ מעברים - shift – העברת מילה מהבאפר למחסנית 

הוספת קשת מהמילה השניה לעליונה - Right ard

והוצאת העליונה אי אפשר לקבל קשתות חוצות רק היטלי יי ייבי יידרת קשתות ימינה ברגע שאפשר בלי להיפטר **–Eage**i

מהמילה העליונה (reduce) על מנת להימנע מהחלטיות אורקל- קלט- עץ, אוסף קשתות לא סדור פלט- סדרת החלטות שהובילה ליצירת העץ מהקלט נתינת הציונים למעברים היא בעיית סיווג שניתן לבצע ע"י חילוץ פיצ'רים מתוך המערכת ואז לתת ציון לכל פעולה מחזיקים את K מחזיקים את-Beam search

כה, בכל צעד מתייחסים לכל העלומה ומעדכנים את מה ניתוח תלויות בגרף- בעל תוצאות טובות יותר מאשר ניתוח תלויות במעברים. נעבור על כל העצים האפשריים וניקח את העץ בעל הציון המקסימלי (סכום הקשתות). מסתכל על עץ שלם ונותן ציון כולל, הנחה- ציון קשת לא תלוי בקשתות האחרות. נגדיר ציוו לכל קשת. לכל זוג צמתים, לכל הכיוונים, ולכל סוג קשת אפשרית וגם לכל

מציאת עץ פורש מקסימלי- ניתן ציון לכל הקשתות נמצא לכל צומת את הקשת המקסימלית הנכנסת אליו.

צומת כשורש

- אם יש מעגל: נפחית מכל קשת הנכנסת לצומת את
- הערך המקסימלי שנכנס לצומת נכווץ את המעגל לצומת בודד ונקרא .2
- לאלגוריתם על העץ החדש
- . נרחיב מחדש את העץ ונבחר את הקשת לפי העץ המוקטן

# Word embeddings

הגדרה: וקטור השיכונים של מילה מסוימת, במרחב הוקטורי.

פוליסמיה: מילה אחת הרבה משמעויות (כוכב. עכבר). מילה אחת. הקשרים שונים. וקטור ממוצע של וקטורים רחוקים. מילים נרדפות: כמה מילים, אותה

משמעות (ירח, לבנה). כמה מילים, אותם הקשרים, וקטורים קרובים. מילים דומות: מתארות חפצים או רעיונות

דומים בעולם (כלב/חתול). כמה מילים הקשרים דומים. וקטורים קרובים עם קשר אלגברי.

מילים קשורות: מתארות חפצים או רעיונות הקשורים לאותו שדה סמנטי (כלב/מלונה). מילים בהקשר זו של זו. וקטורים עם קואורדינטות מסוימות דומות. רשתות סמנטיות: "מילון מרושת". מילים (senses) ממופות למובנים (lemmas) והמובנים מקושרים ביניהם בצורת גרף.

.WordNet

# (Semantic Nets) רשתות סמנטיות Cat lives in → Water ⊭

#### האם אוסף השיכונים באמת מייצג יחס ?דמיון לשוני אמין

1. ניתן לשאול אנשים עד כמה המילים . דומות. זה סובייקטיבי.

2. נשתמש בדמיון קוסינוס

$$cosine(\mathbf{v}, \mathbf{w}) = \frac{\mathbf{v} \cdot \mathbf{w}}{|\mathbf{v}||\mathbf{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2} \sqrt{\sum_{i=1}^{N} w_i^2}}$$

מטריצת מילים/מסמכים: נייצג כל מילה לפי אוסף המסמכים שהיא מופיעה בו כמה בכל מסמר.

> חסרונות של המטריצה הנ"ל: 1. וקטורים ארוכים לכל מילה.

2.מילה כמו The תהיה דומה לשאר המילים הfunctionable ויהיה קשה ללמוד עליהן משהו.

מטריצת שכנויות- נספור הופעות של זוגות מילים

	, ז ווויקשו	ו אוונדו	אנ <i>ו</i> עניינו	117.17	באות מסמן ש המיזה			
	computer	data	result	pie	sugar	count(w)		
cherry	2	8	9	442	25	486		
rawberry	0	0	1	60	19	80		
digital	1670	1683	85	5	4	3447		
formation	3325	3982	378	5	13	7703		
int(context)	4997	5673	473	512	61	11716		
				-				

דוהי נוסחא שעוזרת לנו להתגבר על כך שחלק- PMI מהמילים שכיחות יותר וחלק מהמילים פחות בכך שאנחנו מנרמלים את ההסתברויות <sup>'</sup>

$$PMI(w,c) = \log_2 \frac{P(w,c)}{p(w)p(c)}$$

c ההסתברות ש w תופיע במסמך עם ההקשר- P(w,c)ההסתברות שw תופיע במסמך – P(w)

. ר ההסתברות של ההקשר – *P(c)* . כשהתוצאה תהיה שווה לאפס זה אומר שיש אי תלות בין המילים

רק חיובי) pmi על RELU ביצוע PPMI שיכון ע"י חיזוי הקשרים מקומי -Skip שיכון ע"י חיזוי הקשרים

Gram. נפעיל אלגוריתם חיזוי, אבל נשמור רק את המשקולות הנלמדים.

האלגוריתם: בהינתן מילה w, ננסה לחזות את השכנות שלה c בהסתברות גבוהה ככל הניתן, ונעבור כך על כל המילים בקורפוס. בעצם נחשב מה ההסתברות ש*w,c* מופיעות ביחד .P(+1w.c)

לכל מילה יהיה וקטור מטרה בתפקידה כחוזה, ווקטור הקשר בתפקידה כנחזית.

היפר פרמטרים- רוחב החלון (בכל פעם נדגום רק "חלון" בסביבת המילה וכך נגדיר מה יהיו מילות ההקשר) , כמות הדוגמאות השליליות לכל דוגמה חיובית , מימד הוקטור, נדגום את המילים השליליות מתוך התפלגות שמוסחת לטובת מילים נדירות, אלפא = 0.75 זה ערך

שמוסחת לטובת מילים נדירות, אלפא 
$$P_{lpha}(c) = rac{count(c)^{lpha}}{\sum_{c} count(c)^{lpha}}$$
יעד החיזוי: נשתמש ברגרסיה לוגי $\mu$ יעד החיזוי: נשתמש ברגרסיה לוגי $\mu$ יעד מינו לחשב את ההסתרבות של

יעד החיזוי: נשתמש ברגרסיה לוגיסטית על מנת לחשב את ההסתבבות של על כל דוג' חיובית ניקח כמה p(c/w)שליליות:

$$P(+|w,c_{pos}) \dot{\prod} (P(-|w,c_{neg}))$$

סקיפגראם- הפסד:

 $L_{CE} = -[log\sigma(c_{pos} * w) + \sum_{i=1}^{k} log\sigma(-c_{neg_i} * w)$ 

### סקיפגראם- גרדיאנט:

$$\begin{split} \frac{\partial L_{CE}}{\partial c_{pos}} &= \left[\sigma(c_{pos} * w) - 1\right] w \\ \frac{\partial L_{CE}}{\partial c_{neg}} &= \left[\sigma(c_{neg} * w)\right] w \\ \frac{\partial L_{CE}}{\partial w} &= \left[\sigma(c_{pos} * w) - 1\right] c_{pos} + \sum_{k=0}^{\kappa} \left[\sigma(c_{neg_k} * w)\right] c_{neg_k} \end{split}$$

#### :סקיפגראם- עדכון

 $w^{t+1} = w^t - \eta [[\sigma(c_{pos} * w^t) - 1]c_{pos} + \sum_{i=1}^{k} [\sigma(c_{neg_i} * w)]c_{neg_i}]$ PCA הצגה של וקטורים במימד נמוך ע"י

או ע"י שיכון שכנים סטוכאסטי לפי t (t-SNE) התפלגות

# מטריקות לשערוך מודלי דימיון:

:Pearson correlation

אפשר לכווץ ולהרחיב  $ho X, Y = \frac{cov(X,Y)}{r}$ את התחום. רגיש לפיזור הערכים השונים Xבטווח של

### :Spearman correlation

$$r_s = \frac{cov(R(X), R(Y))}{\sigma R(X)\sigma R(Y)}$$

הרשימות ad כל אחת מהרשימות – F סמנטיקה גבוהה- ייצוג משמעות של ביטוי באמצעות הרכבת חלקיו.

(קשור לAGENT,THEME וכו'.) 3 גישות לסימון תפקידים תמטיים:

1. VerbNet: כתובים שמות התפקידים התמטיים.

> 2. PropBank: רק מספור של הארגומנטים, שמירה על הסדר.

FrameNet.3: כתובים שמות התפקידים

<u>רובוטריקים</u>

1. השוואה בין הווקטורים בקלט ומתן ציון

2. נרמול הציונים שקיבלנו ע"י שימוש

כל מילה שואלת כמה היא צריכה לקחת מכל מילה

:self-attentuina השלבים

בsoftmax (הסתברויות).

3. חישוב הפלט הנוכחי.

 $\mathbf{y}_i = \sum_{j \leq i} \alpha_{ij} \mathbf{x}_j$ 

-המילה כמרכיב של שכבה הבאה Value

לדמיוו ביניהם.

ברמת הייצוג הבאה Query -המילה המתשאלת,

המילה כגורם השוואה – Ke

הספציפיים.

Asha AGENT ARG2: entity given to RECIPIENT ARG1: thing given ARG0: giver DONOR THEME Asha VerbNet AGENT PropBank ARG0: teacher RECIPIENT ARG1: subject TEACHER STUDENT



קלט: רצף מילים או משהו שכולל רצף

3. ביטוי בהטיה מורפולוגית (זמן וגוף)

מודל ה"ערוץ הרועש": בהינתן המשפט

ה"מורעש" מהו ה"מקור" שאותו הרעישו

מילים כמו תמונות או קולות.

פלט: רצף מילים כלשהו.

קשיים במעבר ביו שפות: 1. סדר המילים שונה.

2. אי תאימות במשמעות.

4. אלמנטים חסרים (כינוי גוף)

text->syntax- :משולש ווקווא

בשפת היעד.  $w^* = \arg \max_{w} P(w|a)$ 

 $= \arg \max_{\mathbf{w}} \frac{P(\mathbf{a}|\mathbf{w})}{P(\mathbf{w})} / P(\mathbf{a})$ 

מודל תרגום ברמת המילה:

 $=\arg\max_{\mathbf{w}}P(a|\mathbf{w})P(\mathbf{w})$ 

 $(s), A \mid w^{(t)}) = \prod_{m=0}^{M^{(t)}} p(w_m^{(s)}, a_m \mid w_{a_m}^{(t)}, m, M^{(s)}, M^{(t)})$ 

>semantics->interlingua

לעומת מילות עזר.

מטריקות לבדיקת חילול טקטס:

FrameNet TE/
נאמן מודל תרגום
-Backtranslation משפת היעד לשפת המקור, ואז נשתמש במה שיצרנו בתור דאטה לאימון לתרגום מהמקור ליעד. דוגמה ללמידה חצי

בייקיה. 1.גישת החפיפה – נחליט שיש טקסט מושלם אחד ונשווה בין הרצפים שלו ושל תרגום ( reference

מפוקחת

שנבירם שוה ביו היו האברים שוה המערכת (אפשר עם רצפי תווים או אתיות (CHRF2,2 אפשר עם רצפי תווים או אתיות (I (CHRF2,2 אפשר עם רצפי תווים או אתיות (I (Tunigrams) bigrams that match: wiit ne es ss th he ep pa as st t, (I3 bigrams)

unigram P: 17/17 = 1 unigram R: 17/18 = .944 bigram P: 13/16 = .813 bigram R: 13/17 = .765

REF: witness for the past, HYP1: witness of the past

chrP = (17/17+13/16)/2 = .906 chrR = (17/18+13/17)/2 = .855

חלקי מה שיצא בחיזוי – *P* 

2.מרחב- נלמד מודל שפה גדול ונשווה בין המשפטים ע"י

(reference) מה שהואם חלקי מה שהיה במקור – R  $SelfAttn\left(Q,K,V
ight)=softmax\left(rac{Q\cdot K^{T}}{\sqrt{d_{k}}}
ight)\cdot V$ 

 $y_3$  בהתאמה, נקבל לדוגמא עבור  $y_1,y_2,y_3$  בהתאמה, נקבל לדוגמא עבור בהינחן 3 מילי ב $y_3=\sum_{i=1}^3q(x_3)\cdot k(x_i)^T\cdot v(x_i)$ 

 $dim(d*d_k)$  - W  $Q = X \cdot W_q$  $dim (d*d_k) - W_k$  $dim(d*d) - W_i$ 

 $K = X \cdot W_k$  $V = X \cdot W_n$ 

 $q(\vec{x_i}) = \vec{x_i} \cdot W$  $k(\vec{x_i}) = \vec{x_i} \cdot W_i$  $v\left(\vec{x_i}\right) = \vec{x_i} \cdot W_i$ 



הקשרים השיוריים נותנים לשכבות העליונות גישה ישירה למידע המגיע מהשכבות הנמוכות.

z = LayerNorm(x + SelfAttn(x)) y = LayerNorm(z + FFNN(z))

בנורמליזציה נחשב:

$$\mu = \frac{1}{d_h} \sum_{i=1}^{d_h} x_i$$

$$\sigma = \sqrt{\frac{1}{d_h} \sum_{i=1}^{d_h} (x_i - \mu)^2}$$

Multihead צומי מרובה ראשים

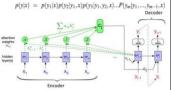
self- כמה שכבות של :Attention attention שנקראות ראשים. לכל ראש יש סט מטריצות משלו. כל ראש יכול למצוא קשתות שונות בין המילים במשפט.

נוסיף לשכבת הצומי מיקום עם positional embeddings

למידה בהעברה: למידת מערכת למטרה מסוימת ושימוש באלמנטים הנלמדים שלה למשימה אחרת. (skip-gram) יעדכון הפרמטרים שנלמדו :Fine-tuning למשימה המקורית על גבי המשימה החדשה.

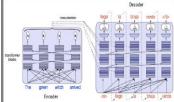
( seq to seq) חילול טקסט

ציוני מודל השפה הגדול (משפט למשפט או מילה למילה לפי יישור) מקודד (encoder)- לוקח טקסט (או אובייקט אחר) מפענח (או כמה) לוקח וקטור (decoder ') מפענח  $p(y|x) = p(y_1|x)p(y_2|y_1,x)p(y_3|y_1,y_2,x)...P(y_m|y_1,...,y_{m-1},x)$ 



מקודד רובוטריק -כמו קודם אבל יכול לראות את כל יבל אבל הסתכל רק אחורה בפלט, אבל יכול להסתכל הק

cross-attention כול להסתכל על כל הקלט בעזרת



הלמידה כאו באמצעות קרוס אנטרופי רגיל- מינוס לוג הסתברות של המילה ,ונכריח את המודל בכל שלב לקחת את המילה הבאה האמיתית / חיפוש אלומה. שני סוגים של מודלים ברובוטריק:

autoregressive מודל שפה-מסתכל רק אחורה. למידה מעוברת

2.מודל חיזוי או מקודד לבעיות תרגום או ממוסך (מסתכל מודל שפה דו כיווני ELMO: חיבור של מודל קדימה

מודל אחורה רובוטריק כמודל שפה דו כיווני. נסתיר לחלק-

מהמילים את עצמו. יכול לשמש כמקודד תרגום. חיזוי קורפרנס- איזכורים של אותה ישות יוצרים cluster אם נמייו לפי סדר הופעה נקבל שרשרת לכל איזכור חדש נחשב הסתברות - Mention pairs הישור עבור כל אזכור קודם (האם הם קשורים לאותה שות) – זוג לאזכור החדש עם כל אזכור קודם נבדוק עבור האזכור את כל האזכורים - Mention rank

לפניו ונבחר את זה עם ההסתברות המקסימלית, האזכור

, המשפט הראשון הוא הגרעין והשני לווין דני הלך לים, הוא אוהב תות – קוהסיבי ולא קוהרנטי **קורפרנס coreference** - ביטויים . שמצביעים על אותן ישויות. סוגי אזכורים: (ושמות)

עם הציון הגבוה הוא זה שיש לו הכי הרבה משמעות

ריישויות) שהוא יכול להצטרף אליהן (הקלאסטרים)(=יישויות) שהוא יכול להצטרף אליהן בעזרת שימוש בפיצ'רים תלויי ישות ותלויי שרשרת

vc.her,mb :דוגמאות: אזכורים שכבר ראינו

(she,(vc,her,mb)) - Mention ramk (she,((vc,her),(mb)))— Entity based

למצוא קשר ביו שני משפטים עוקבים

(she,her),(she,vc) , ( she,mb)- Mention pairs

לכידות טקסט- קוהרנטי -טקסט הוא קוהרנטי אם ניתן

קור בין פני מספט ליוב ב קורסיבי- טקסט הוא קורסיבי אם יש נושא משותף דני הלך לים, הוא אוהב לשחות -קורסיבי וקוררנטי,

she :האזכור החדש

עבור הבנת האזכור הקיים (barak Obama, president) עבור הבנת האזכור הקיים – Entity based

1. צירופים שמניים בלתי מיודעים: קראתי אתמול כתבה מעניינת.

2. צירופים שמניים מיודעים: לחצתי על קישור לכתבה במקרה. 

4. אזכור ריק: יובל קרא את הכתבה ואז . [] הלך הביתה.

# מטריקות:

בעיה קשה. האם מעניינים אותנו האזכורים, הישויות, או הקשרים שבין הישויות? נמדוד **קשרים**: לכל אזכור, כמה מבין הקשרים לשאר אזכורי הישות נמצאו ע"י המערכת? כנגה מערכת שמקשרת כל אזכור לזה שקדם ל

שרשרת יחידה) ונקבל תוצאה מאוד גבוהו נמדוד אזכורים: לכל אזכור, כמה מתוך השרשרת שהענקנו נמצאת גם בשרשרת האמיתית?

נבנה מערכת שיוצרת רק יחידונים (singletons) ונקבל ברecision=1 בלי לפגוע ב-ecall נמדוד ישויות (צבירים): האם נמצאו צבירים

דנותו מספיק? קשה למצוא יישור אופטימלי בין צבירים שאינם זהים, או לתת למערכת דרכים לתיקון, במיוחד אם מספר

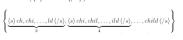
 $chrF2,2 = 5 \frac{chrP*chrR}{4chrP+chrR} = .86$ 

#### *OOV* מילים לא מוכרות

המילה - Character LSTM - נועד לטפל ב OOV בטסט. החסרה בטסט (הOOV) תקבל את הוקטור של <unk> לאחר מכו נוכ לעשות LSTM דו כיווני לחבר את התוצאות . לFFN ולקבל תג מתאים לכל מילה

skip מטפל ב OOV באימון. מתפקד כמו Fast text 3-6 אבל ה subwords הם כל רצפי התווים באורך של המילה הרצויה עם חפיפות

 $subword(\langle s \rangle child\langle /s \rangle) =$ 



 $emb\left( child\right) =emb_{T}\left( child\right) +$ בחלק את המילה לחלקים - Subword tokenization ללא חפיפה. (רוב הטרנספורמרים משתמשים בזה)

> תיאור האלגוריתם:  $\mathbf{BPE}$  אלג $\rightarrow V$ |V| < k כל עוד

.D־ב נפוץ הכי נפוץ ביר הטוקנים הכי בוץ ב- $t_l t_r$  $t_l t_r \rightarrow t_{new}$  $V \cup \{t_{new}\} \rightarrow V$  $t_{new}$ ב  $t_l t_r$  ב' בל מופע של V החזר את

> מערכות שומרות מצבים בצ'טים: זיהוי דיבור אוטומטי->הבנת השפה

-המדוברת->מעקב אחר מצב השיחה -מדיניות הדיאלוג-> ייצור שפה טבעית->טקסט לדיבור וחוזר חלילה.

מידול שיח- סכימת PDTB: Temporal: קשר סינכרוני, זמן. :Contingency קשר סיבתי. Comparison: קשר ניגודי. Expansion: קשר הוספה.

המשר אמבדינגס:

ווים אילתה באחזור מידע, מחפשים שאילתה – TF-idf מנוע חיפוש ורוצים לדעת איזה מסמך להחזיר

 $tf-idf_1(t,d)=log(1+f_{t,d})-log(n_t/N)$ (מילה מתוך השאילתה) t סמות הפעמים שהמילה  $f_{t,d}$ 

d ופיעה במסמר בהם t כמות המסמכים שהמילה t הופיעה בהם  $n_i$ N כמות המסמכים

החיסור נועד לתת משמעות למילים נדירות