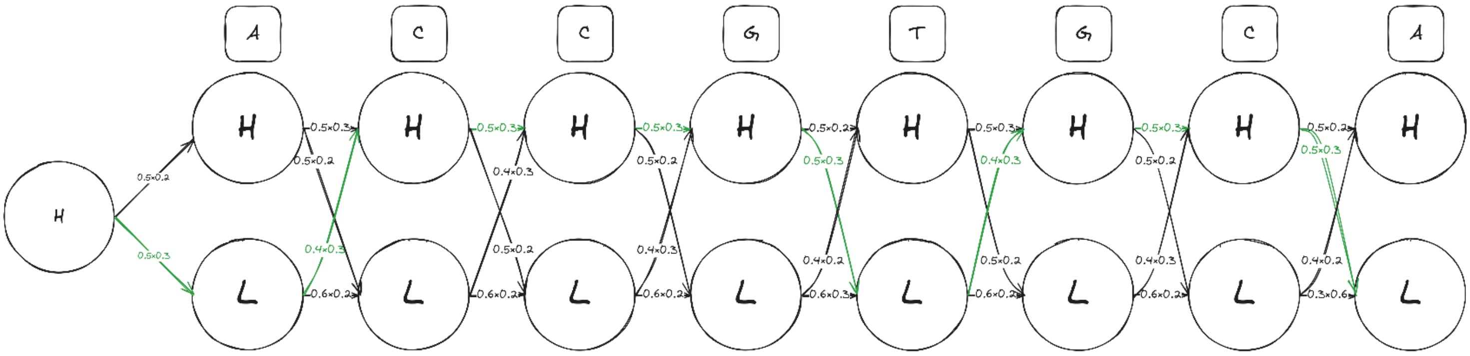
Avraham Asraf : 315774570

Schnaidman Elchanan: 316092436

עיבוד שפה טבעית

1. לאחר הפעלת אלגוריתם ויטרבי נקבל את המסלול הבא:LHHHLHHL

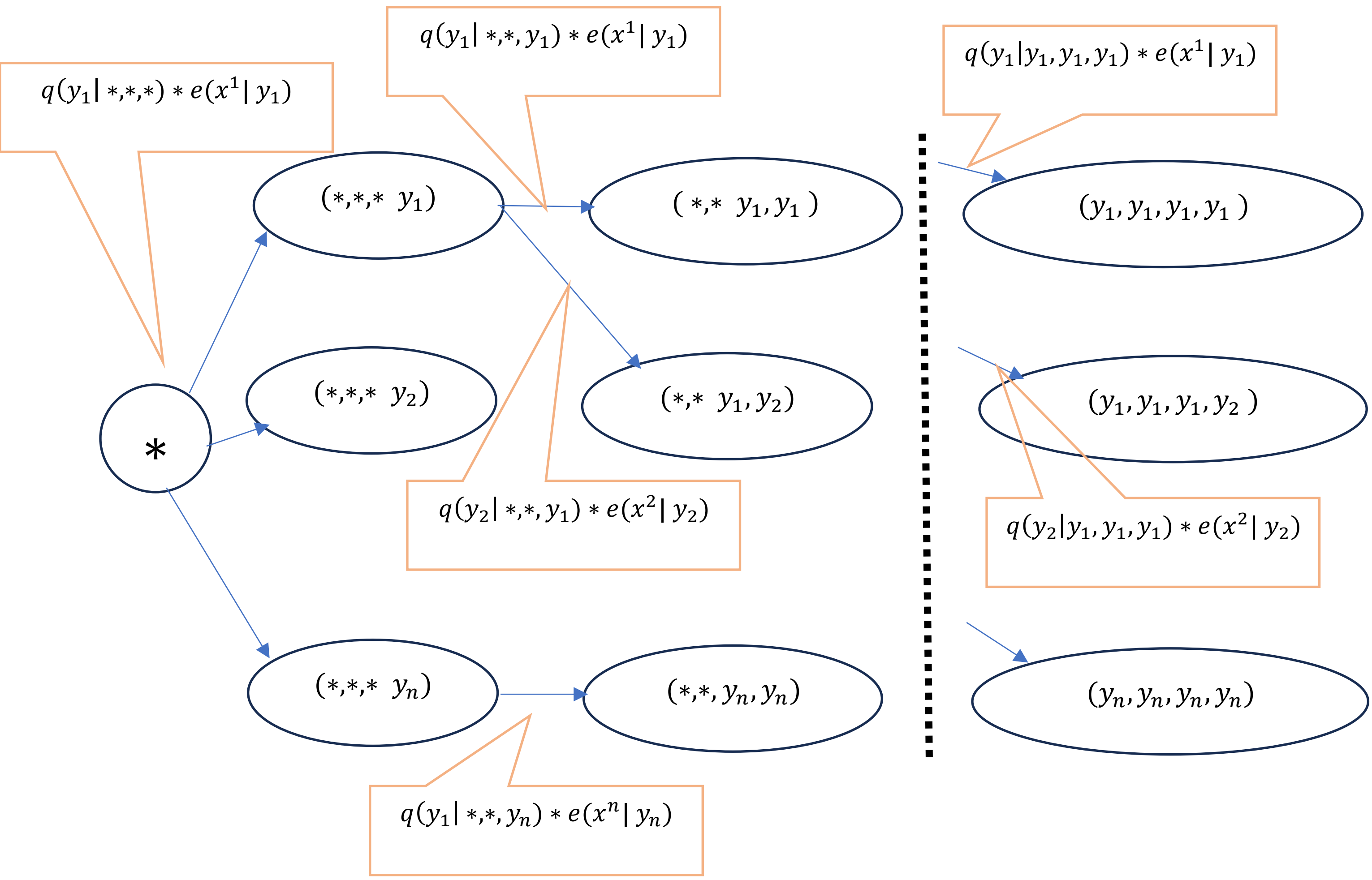
בגרף המצורף מתואר האלגוריתם, המסלול המקסימלי מצורף בירוק:



H	1	*:1×1×0.2	L:0.3×0.3×0.4 =0.036	H:0.036×0.5×0.3 =0.0054	H:0.0054×0.5×0.3 =0.00081	H:0.00081×0.5×0.2 =0.000081	L:0.0001215×0.12 =0.00001458	H:0.00001458×0.15 =0.000002187	H:0.000002187×0.1 =0.0000002187
L		*:1×1×0.3	L:0.3×0.6×0.2 =0.036	L:0.036×0.2×0.6 =0.00432	H:0.0054×0.1 =0.00054	H:0.00081×0.15 =0.0001215	L:0.0001215×0.12 =0.00001458	L:0.00001458×0.12 =0.0000017496	H:0.000002187×0.15 =0.00000032805
	0	1	2	3	4	5	6	7	9

2. נצייר איך זה נראה בתצורה גרפית קודם:

נניח שיש $\max_i e(x_i|y_1) = k$, כלומר x_k ממקסם את הסתברות ה-emission עבור y_1. נסמן את זה x^1. כלומר, ה-x שממקסם את הסתברות ה-emission עבור y_1 יהיה x^1 ועבור y_2 יהיה x^2 וכן הלאה.



הרציונל שעומד מאחורי לקיחת רק קשתות עם משקלים $e(x^i| y_i)$ ולא להוסיף מצבים וקשתות עם $e(x^i| y_j), i \neq j$, זה מפני (אם נסתכל רק על המעבר השני, בה"כ) יהיו לנו 2 מצבים שהם זהים אך עם מעברי משקלים שונים אליהם: $(*,*, y_1, y_1)$ וקשת אחת מ- $(*,*,*, y_1)$ עוברת עם המשקל $q(y_1| *,*, y_1) * e(x^1| y_1)$ אל $(*,*, y_1, y_1)_1$ **הראשון** וקשת נוספת מאותו מקור אל $(*,*, y_1, y_1)_2$ **השני** עם משקל $q(y_1| *,*, y_1) * e(x^2| y_1)$.

נניח בשלילה שקיים מסלול אופטימלי שעובר דרך $(*,*, y_1, y_1)_2$. מכיוון שמעברי ההמשך אחרי $(*,*, y_1, y_1)_2$ תלויים במצב $(*,*, y_1, y_1)$ (כי המצב שהוא מייצג את התג הנוכחי ועוד 3 תגי עבר הוא זהה), לכן נוכל לבנות גם מסלול שעובר את אותה דרך עד $(*,*, y_1, y_1)_2$ אך ממשיך ל- $(*,*, y_1, y_1)_1$ ואז ממשיך למצבים מהמסלול האופטימלי של ההנחה.

מכיוון שלפי הגדרה שלנו $e(x^1| y_1) > e(x^2| y_1)$ אזי

$q(y_1 | *,*, y_1) * e(x^1 | y_1) > q(y_1 | *,*, y_1) * e(x^2 | y_1)$ שזה אומר שהמשקל של הקשת שבחרנו להגעה למצב $(*,*, y_1, y_1)_1$ הוא יותר גדול. את הקשת הבאה לחזור למסלול האופטימלי היא תהיה עם אותו משקל כי הסתברות ה-*emission* נוכל לבחור זהה והסתברות המעבר זהה כי המצבים הקודמים זהים (מה שהשתנה זה הסתברות ה-*emission*.

קיבלנו מסלול עם הסתברות יותר אופטימלית, לכן בהכרח נוכל להוריד מצבים עם הסתברות *emission* אליהם מהצורה $e(x^i | y_j), i \neq j$.

אזי, קיבלנו את הייצוג הגרפי הנ"ל ובעצם נצטרך להריץ ויטרבי עד שנאסוף מספיק מילים שנגיע ל-n. (כלומר, n איטרציות).

אם כן, האלגוריתם יהיה באופן הבא:

Input: an integer n, parameters $q(w|t, u, v)$ and $e(x|s)$.

Definitions: Define k to be the set of possible tags. Define $k_{-2} = k_{-1} = k_0 = \{*\}$, and $k_k = k$ for $k = 1 \cdots n$. Define V to be the set of possible words.

Initialization: set $\pi(0,*,*,*) = 1$

Algorithm:

- For $k = 1 \dots n$:
  - For $t \in k_{k-2}, u \in k_{k-1}, v \in k_k$:
    $\pi(k, t, u, v) = \max_{w \in s_{k-3}} (\pi(k-1, w, t, u) * q(v \mid w, t, u) * e(x^v | v)$

Return: $\max_{t \in k_{n-2},\ u \in k_{n-1},\ v \in k_n} (\pi(n, t, u, v) * q(STOP \mid t, u, v)$

תוצאות קטע הקוד:

```
import data base
(b) Implementation of the most likely tag baseline:
        The error rate for known words is    0.165
        The error rate for unknown words is  0.741
        The total error rate is  0.325

(c) Implementation of a bigram HMM tagger:
        The error rate for known words is    0.13
        The error rate for unknown words is  0.765
        The total error rate is              0.199

(d) Using Add-one smoothing
        The error rate for known words is    0.15
        The error rate for unknown words is  0.743
        The total error rate is              0.215

(e) Using pseudo-words
        The error rate for known words is    0.129
        The error rate for unknown words is  0.684
        The total error rate is              0.195

(d) Using Add-one smoothing and pseudo-words
        The error rate for known words is    0.146
        The error rate for unknown words is  0.655
        The total error rate is              0.206
```

המגמות אותם ניתן לראות:

1. מודל הHMM משפר משמעותית את החיזוי.
2. add-one-smooting משפר את החיזוי עבור מילים שלא ניראו, אך מוריד את הדיוק עבור מילים שכבר נראו, ואף באופן כללי.
3. pseudo-words משפרים באופן משמעותי יותר את השגיאה עבור מילים שלא ראינו, אך השפעתם על המילים שכבר ראינו היא אפסית. זאת גם הגישה שמשיגה את השגיאה הכללית הנמוכה ביותר.
4. שימוש גם ב Add-one smoothing וגם pseudo-words משיג את התוצאה הטובה ביותר עבור מילים שלא ניראו, אך לא את באופן כללי.

נעיר שקיימות טכניקות בסיסיות שהיו יכולות לשפר משמעותית את המודל עבור מילים שלא קיימות באוצר המילים, כמו למשל – הורדת תחילית וסופית של מילה, חיפוש שלא תלוי בדיקת כל המילים בlowercase. אך אנחנו נשארנו במסגרת הטכניקות שהוצגו בתרגיל.