שאלה 1

שאלה 1.1: דיוק ה-parsing ה-dummy parser עבור משפטים באורך לכל היותר-40 מילים:

Bracketing Recall = 0.00

Bracketing Precision = 0.00

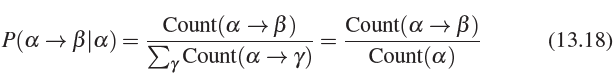
Bracketing FMeasure = -1.#J

שאלה 1.2: דיוק התיוג של ה-dummy parser עבור משפטים באורך לכל היותר 40 מילים:

Tagging accuracy = 18.78

שאלה 2

שאלה 2.1: הנוסחה לחישוב ההסתברויות לחוק (פרק 13 עמ' 8 בספר הלימוד):



שאלה 2.2: פסאודו קוד לבינרזיציה של עצים: (קוד רקורסיבי, הקריאה הראשונה מעבירה כפרמטר את שורש העץ)

**CFG\_TO\_CNF**(Input: ***node***)

1. If ***node*** has daughters:
   1. Invoke CFG\_TO\_CNF recursively on first (leftmost) daughter
   2. If there are more than two daughters:
      1. Create a new artificial node ***artificialNode*** (for node identifier scheme, see bellow)
      2. Remove redundant daughters from ***node*** and assign all “redundant” daughters to ***artificialNode***
   3. If (at this point) ***node*** has two daughters: (see explanation bellow)
      1. Invoke CFG\_TO\_CNF recursively on second (rightmost) daughter

הסבר לשלב 1.3: בשלב זה באלגוריתם, אחת משתי האפשרויות מתקיימת:

* ל-***node*** היו לכתחילה בת אחת או שתי בנות: במצב הזה, לא רץ שלב 1.2, ובשלב 1.3 המצב נותר כשהיה, ופשוט עובר על הבנים (במידה וקיימים) של הnode ושולח אותם כדי לבצע עליהם המרה ל-CNF.
* במידה ול-***node*** היו לכתחילה יותר משתי בנות: במצב הזה, שלב 1.2 רץ. בשלב 1.2 יוצרים בת "חדשה מלאכותית", וכל שאר הבנות (החל מהאינדקס השני, ועד לסוף רשימת הבנות), הופכים להיות בנות של הצומת המלאכותי. כלומר, שוב ל-***node*** יש שתי בנות - הבת השמאלית המקורית והבת הימנית המלאכותית החדשה.
  + במקרה הזה, נוספים ***n - 2*** בנות ל-***node*** (מספר הבנות הוא ***n***), שכן עבור כל בת (חוץ מהשתיים האחרונות), מוסיפים ***node*** ל-CYK matrix.

שאלה 2.2:

אלגוריתם הבינריזציה נדרש עבור הרצת אלגוריתם ה-CYK, שכן עבור CYK החוקים נדרשים להיות במבנה CNF. (במימוש שלנו, הוספנו לאלגוריתם תמיכה בחוקים אונאריים).

שאלה 2.3:

* כאשר מבצעים בינריזציה של העצים, כל צומת עם יותר מאח אחד מאבד מה"קונטקסט" שלו – הוא לא זוכר אילו אחים היו לו. כיוון שלקונטקסט יש השפעה בשפה טבעית, איבוד מוחלט של הקונטקסט פוגע בביצועים.

כדי לבצע מרקובזציה הוספנו עבור כל node משתנה נוסף, m\_lstBrothers שמייצג את כל האחים השמאליים הקיימים עבור הnode.

מה שנוסף לפסאודו קוד הוא שעבור בינארזציה אנחנו מבקשים גם h שמסמן כמה בנים שמאלים לקחת, וככה עבור כל node שעובר להיות בCNF אנחנו מעדכנים את הm\_sIdentifier להיות עם שרשור של h האחים השמאלים של אותו הnode.

**שאלה 3**

**Pre-processing**

1. Save map of lexical rules by the product word
2. Save map of unary grammar rules, by RHS argument
3. Save map of binary grammar rules, by leftmost RHS argument
4. Create the right upper-half of a matrix and populate it with empty maps (for holding different symbols probabilities, the matrix is indexed like Figure 13.4 (ch. 13, pp. 8 in the textbook)
5. Create a map for holding the backtrace (for reconstructing the probable tree)
6. Save ***default-prob*** the average probability of NN tag (for assigning to unknown words)

**PROBABILISTIC-CYK**(Input: ***sentence***)

1. cyk-matrix ← new “half matrix”[[1]](#footnote-1) of length LEN(***sentence***) // init
2. cyk-backtrace ← new empty map // init
3. For j ← 1 to LEN(***sentence***) // external loop – iterate over words in sentence
   1. If ***sentence***[j-1] is an unknown word
      1. cykmatrix[j-1, j, ‘NN’] ← ***default-prob***
      2. cyk-backtrace[j-1, j, ‘NN’] ← (-1, ***sentence***[j-1], null) // -1 marks the value triplet as a terminal symbol)
   2. Else:
      1. Foreach LHS of lexical rule {A | A 🡪 ***sentence***[j-1]}
         1. cykmatrix[j-1, j, A] ← P(A 🡪 ***sentence***[j-1])
         2. cyk-backtrace[j-1, j, A] ← (-1, ***sentence***[j-1], null) // -1 marks the value triplet as a terminal symbol)
   3. Repeat till convergence or max iteration limit reached:
      1. For each unary rule which it’s RHS is in {A | (j-1,j,A) ∈ cyk-matrix}: // this search is optimized through indexing in the pre-processing stage
         1. If no previous probability for A exists:
            1. cyk-matrix[j-1, j, A] ← (-2, A null) // -2 marks the value triplet as a unary production
   4. for i ← j – 2 down to 0 do:
      1. for k ← i + 1 to j-1 do:

שאלה 4:

פסאודו קוד עבור השחזור:

עבור h=0:

Bracketing Precision = 44.47

Bracketing FMeasure = 33.14

Complete match = 0.00

Average crossing = 3.98

No crossing = 21.33

2 or less crossing = 43.78

Tagging accuracy = 79.54

עבור h=1:

\_\_\_\_\_\_\_\_\_\_\_\_ len<=40 \_\_\_\_\_\_\_\_\_\_\_\_\_

Number of sentence = 900

Bracketing Recall = 26.32

Bracketing Precision = 43.73

Bracketing FMeasure = 32.86

Complete match = 0.00

Average crossing = 4.04

No crossing = 21.89

2 or less crossing = 43.67

Tagging accuracy = 77.89

עבור h=2:

\_\_\_\_\_\_\_\_\_\_\_\_ len<=40 \_\_\_\_\_\_\_\_\_\_\_\_\_

Number of sentence = 1350

Bracketing Recall = 26.69

Bracketing Precision = 43.19

Bracketing FMeasure = 32.99

Complete match = 0.00

Average crossing = 4.19

No crossing = 21.48

2 or less crossing = 42.15

Tagging accuracy = 77.21

עבור h=-1:

\_\_\_\_\_\_\_\_\_\_\_\_ len<=40 \_\_\_\_\_\_\_\_\_\_\_\_\_

Number of sentence = 1800

Bracketing Recall = 26.90

Bracketing Precision = 42.84

Bracketing FMeasure = 33.05

Complete match = 0.00

Average crossing = 4.28

No crossing = 21.33

2 or less crossing = 41.67

Tagging accuracy = 76.88

הסתכלנו על ה10 המשפטים הראשונים עבור h=0 (התוצאה הטובה ביותר):

סווג טעויות מהעלים של העץ לכיוון השורש:

· יש מספר מילים(סגמנטים)

שאלה 5:

השתמשנו באולגוריתם parent encoding - מוסיפים את האבא לכל אחד מהבנים, ככה שהחוקים מודיעים גם למי האבא, ואז זה ממש דומה לbi-gram(כמו שעשינו בממן 12).

Sentence #10: **IFRAL ARD yyCM SMNKUL H BIJWX H LAWMI yyCM AMR KI MMLA MQWM FR H EBWDH W H RWWXH yyCM DWD MGN yyCM HQIM WEDH BIN yyDASH MFRDIT yyCM F HMLICH LHGDIL B AWPN MFMEWTI AT H QNSWT L H MESIQIM yyDOT**

משפט: **ישראל ארד, סמנכ"ל הביטוח הלאומי, אמר כי ממלא מקום שר העבודה והרווחה, דוד מגן, הקים ועדה בין-משרדית, שהמליצה להגדיל באופן משמעותי את הקנסות למעסיקים.**

השוואה בין הניתוח הנכון של המשפט ("gold parsing"), לבין הניתוח של האלגוריתם: **שגיאה! אובייקט מוטבע לא חוקי.**

1. ראה Figure 13.4 בפרק 13 בספר הקורס (עמ' 8). האינדקסים באלגוריתם מתייחסים גם הם לאינדקסים של התאים בטבלה. [↑](#footnote-ref-1)