

x_2 Shift \rightarrow left \rightarrow Shift $\xrightarrow{x_3}$ left \rightarrow right \rightarrow Shift $\xrightarrow{x_L}$ left $\xrightarrow{-1}$ "KQ"

John Likes balloons that pop and eggs

John Likes balloons that pop and eggs

Buffer

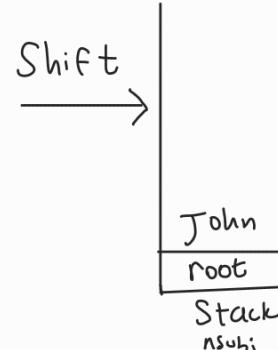
John Likes balloons that pop and eggs

Buffer

~~John~~ Likes balloons that pop and eggs



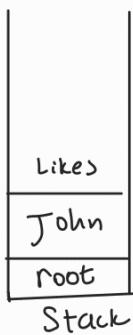
John Likes balloons that pop and eggs



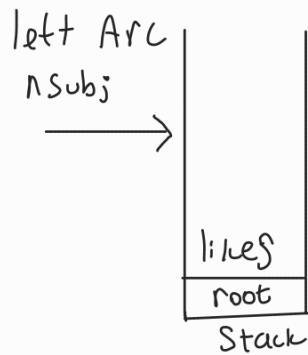
John Likes balloons that pop and eggs

Buffer

~~John Likes balloons that pop and eggs~~



John likes balloons that pop and eggs



ר' נבו ז אל'ז
Shift
→

Buffer

~~John Likes balloons that pop and eggs~~



Buffer

~~John Likes balloons that pop and eggs~~

The diagram shows a stack structure with three horizontal lines representing the stack's boundaries. The stack contains four items from top to bottom: "pop", "ballons", "Likes", and "root". To the left of the stack, there is a large arrow pointing right, above which the word "mark" is written.

```

graph TD
    ACL[ACL] --- nsubj((nsubj))
    ACL --- mark((mark))
    nsubj --- Likes[Likes]
    nsubj --- balloons[balloons]
    mark --- pop[pop]
    mark --- and[and]
    pop --- eggs[eggs]
    and --- eggs
    Likes --- balloons

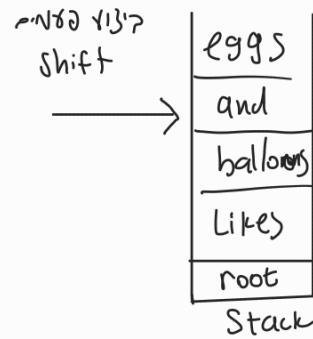
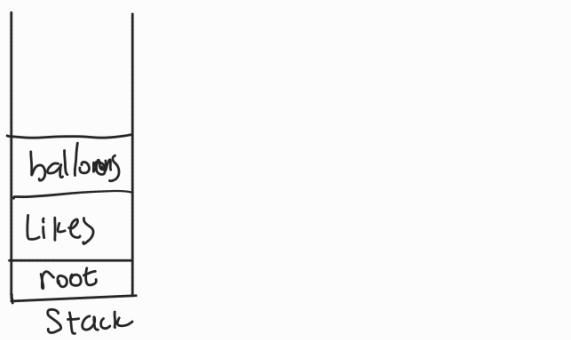
```

Buffer

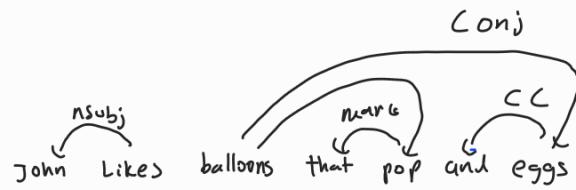
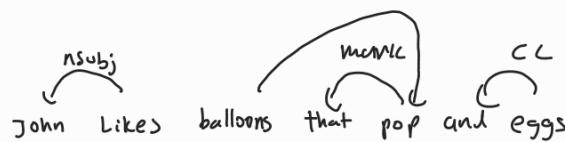
~~John likes balloons that pop and eggs~~

Buffer

~~John Likes balloons that pop and eggs~~



Left Arc
cc



Buffer

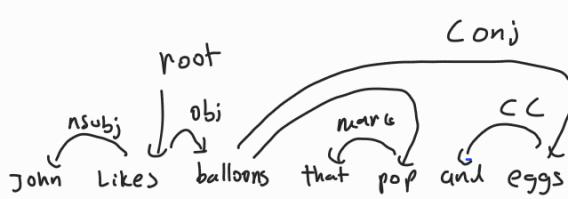
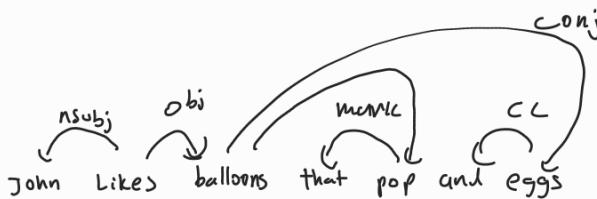
John	Likes	balloons	that	pop	and	eggs
------	-------	----------	------	-----	-----	------



Right Arc
conj



Right Arc
obj

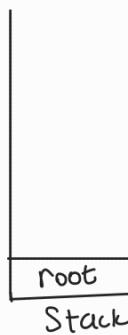


Buffer

John	Likes	balloons	that	pop	and	eggs
------	-------	----------	------	-----	-----	------



Right Arc
root



2 NLCQ

Shift → left → right → right → shift → left
→ right → reduce → shift → left-right

P'383

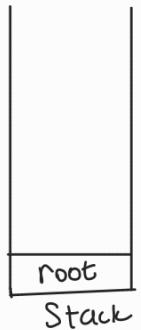
11

JDO

John Likes balloons that pop and eggs

Buffer

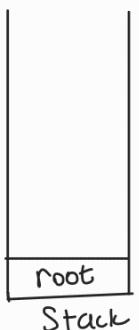
John	Likes	balloons	that	pop	and	eggs
------	-------	----------	------	-----	-----	------



nsubj
 ↙
 John Likes balloons that pop and eggs

Buffer

John	Likes	balloons	that	pop	and	eggs
------	-------	----------	------	-----	-----	------



nsubj
 ↗ root
 obj
 ↙
 John Likes balloons that pop and eggs

Buffer

John	Likes	balloons	that	pop	and	eggs
------	-------	----------	------	-----	-----	------



Shift



nsubj
 ↙
 John Likes balloons that pop and eggs

Buffer

John	Likes	balloons	that	pop	and	eggs
------	-------	----------	------	-----	-----	------

Right Arc
 ↗ root

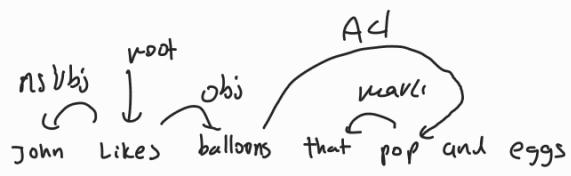
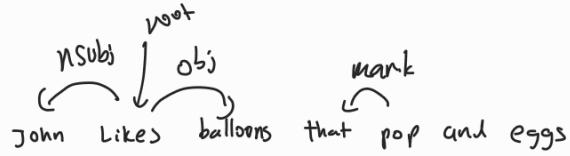
nsubj
 ↗ root
 obj
 ↙
 John Likes balloons that pop and eggs

Buffer

John	Likes	balloons	that	pop	and	eggs
------	-------	----------	------	-----	-----	------

Shift

Right Arc
 ↗ objLeft Arc
 ↗ nanc



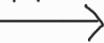
Buffer

John	Likes	balloons	that	pop	and	eggs
------	-------	----------	------	-----	-----	------

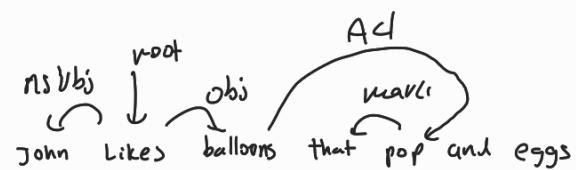
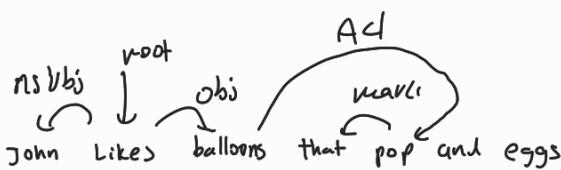
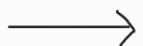


Right Arc

Acl



reduc



Buffer

John	Likes	balloons	that	pop	and	eggs
------	-------	----------	------	-----	-----	------

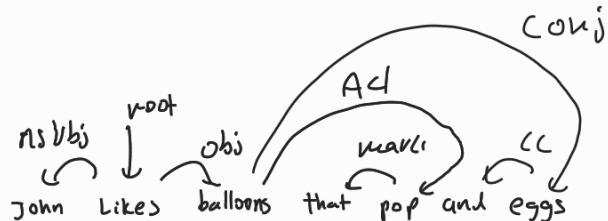
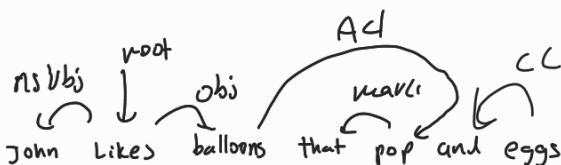
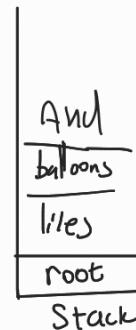


Shift



left Arc
CC

John	Likes	balloons	that	pop	and	eggs
------	-------	----------	------	-----	-----	------



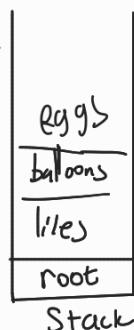
Buffer

John	Likes	balloons	that	pop	and	eggs
------	-------	----------	------	-----	-----	------



Right Arc

conj

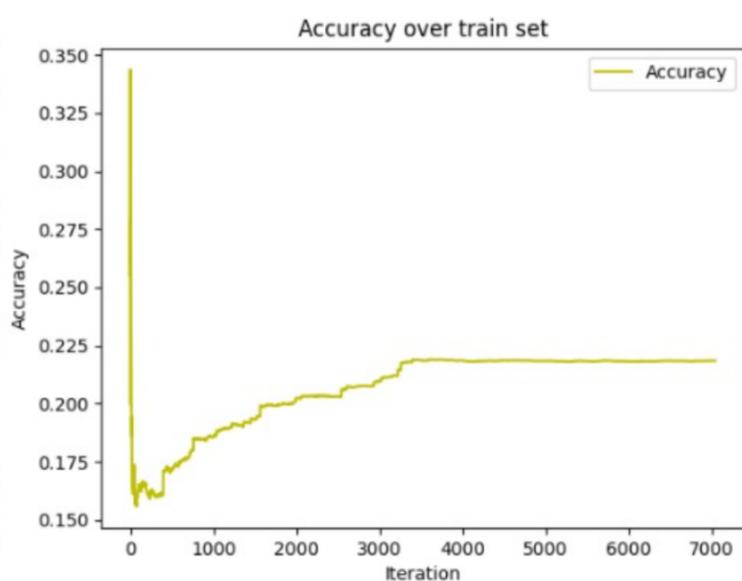


Buffer

John	Likes	balloons	that	pop	and	eggs
------	-------	----------	------	-----	-----	------

The Average Accuracy over All Sentences in the test set was 0.20551485446786158

This is the accuracy over the train set



Ignoring the peak at zero, we can see from the learning curve that 4200 iterations are enough to reach a minimum point. We also notice that although our accuracy increases during training, it stabilizes at a low 0.2. These is indicative of features that are not rich enough.

