

Q1

(a)

i) 그라디언트가 작은 계수값을 가지기 때문에 업데이트 속도는 안정적이게 하는 동시에 빠른 업데이트가 가능하다. 전체적으로는 분산을 줄여 오버피팅을 방지하고 좀 더 빠르게 수렴할 수 있게 한다.

ii)  $\alpha$  와  $\frac{m}{\sqrt{v}}$  의 element wise 곱으로  $\theta$  update.

$$\begin{bmatrix} \sqrt{v} \downarrow & \text{stepsize} \uparrow \\ \uparrow & \downarrow \end{bmatrix}$$

⇒ parameter 가 비슷한 속도로 학습하고, 더 빠르게 수렴하는 데에 도움을 준다.

(b)

$$\therefore \begin{bmatrix} 0 & \text{with prob. } p_{\text{drop}} \\ 1 & \text{" } (1 - p_{\text{drop}}) \end{bmatrix}$$

$$\begin{aligned} \therefore h_i &= E_{p_{\text{drop}}} [h_{\text{drop}}]_i = E_{p_{\text{drop}}} (\gamma d \cdot h)_i \\ &= (1 - p_{\text{drop}}) h_i \cdot \gamma \end{aligned}$$

$$\therefore \gamma = \frac{1}{1 - p_{\text{drop}}}$$

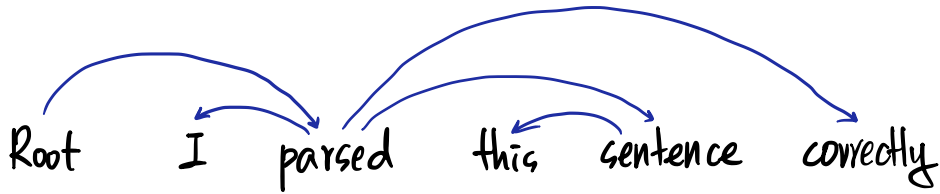
ii) evaluation 은 모델의 성능 평가를 위한 단계이므로 dropout이 필요없다.

또한 dropout을 하게 되면 동일한 input에 대해서도 다른 값을 낼 수 있으므로 쓰지 않는다.

( $\therefore$  dropout은 랜덤)

Q2

(a)



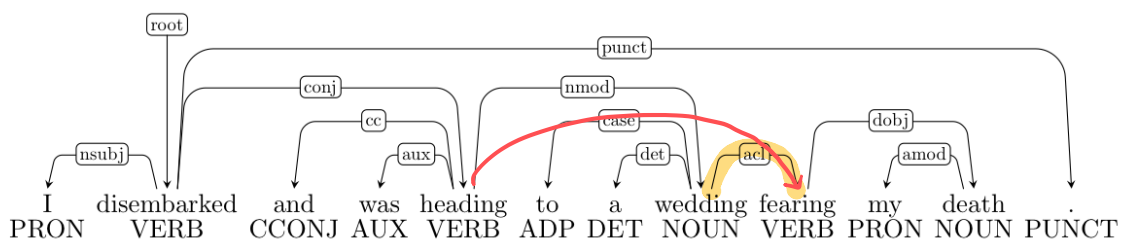
Stack	Buffer	New dependency	Transition
[ Root ]	[ I, parsed, this, sentence, correctly ]		Initial Configuration <sup>0</sup>
[ Root, I ]	[ parsed, ..., correctly ]		Shift
[ Root, I, parsed ]	[ this, ..., correctly ]		Shift
[ Root, parsed ]	"	parsed → I	Left Arc
[ Root, parsed, this ]	[ sentence, correctly ]		Shift
[ Root, parsed, this, sentence ]	[ correctly ]		Shift
[ Root, parsed, sentence ]	"	sentence → this	Left Arc
[ Root, parsed ]	"	parsed → sentence	Right Arc
[ Root, parsed, correctly ]			Shift
[ Root, parsed ]		parsed → correctly	Right Arc
[ Root ]		Root → parsed	

$$n = 5$$

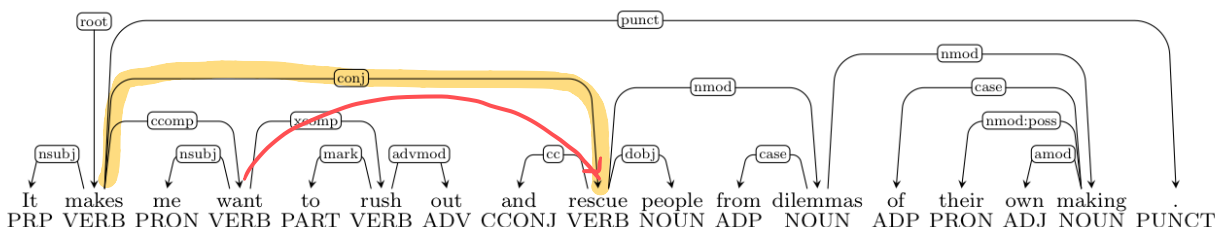
$$\Rightarrow 10 = 2n$$

(b)  $2n$  step.  $\begin{cases} \text{shift or } n \text{ step} \\ \text{Arc } n \text{ step} \end{cases}$

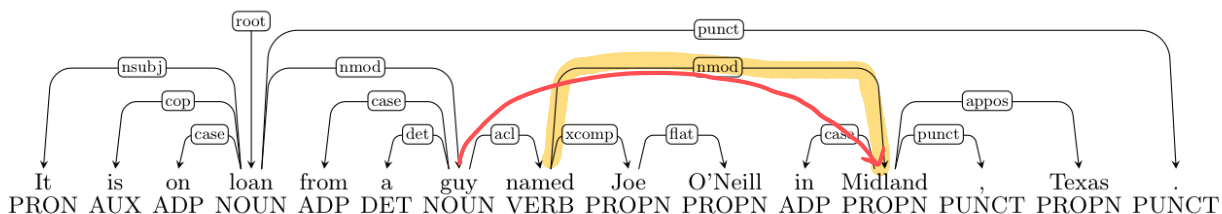
i.



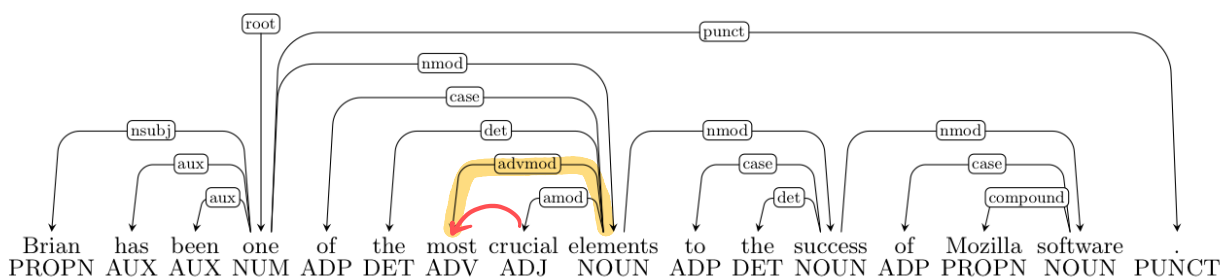
ii.



iii.



iv.



Q	Error Type	Incorrect Dependency	Correct Dependency
i)	Verb Phrase attachment error	wedding → fearing	heading → fearing
ii)	Prepositional Attachment error ✓	makes → rescue	want → rescue
iii)	✓	named → midland	guy → midland
iv)	Modifier Attachment error	element → most	crucial → most