ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Natural Language Processing
Fall 2022

Prof. Ryan Cotterell

Simon Wachter: Assignment 2

siwachte@ethz.ch, 19-920-198

11/12/2022 - 11:22h

Question 1:

a) Prove that the expectation semiring satisfies the semiring axioms:

- $(\mathbb{R} \times \mathbb{R}, \oplus, \mathbf{0})$ must be a commutative monoid with identity element $\mathbf{0}$:

$$\begin{align}
(\langle x, y\rangle \oplus \langle x', y'\rangle) \oplus \langle x'', y''\rangle &= \langle x + x', y + y'\rangle \oplus \langle x'', y''\rangle \tag{1}\\
&= \langle x + x' + x'', y + y' + y''\rangle \tag{2}\\
&= \langle x, y\rangle \oplus \langle x' + x'', y' + y''\rangle \tag{3}\\
&= \langle x, y\rangle \oplus (\langle x', y'\rangle \oplus \langle x'', y''\rangle) \tag{4}
\end{align}$$

$$\begin{align}
\mathbf{0} + \langle x, y\rangle &= \langle 0, 0\rangle \oplus \langle x, y\rangle \tag{5}\\
&= \langle 0 + x, 0 + y\rangle \tag{6}\\
&= \langle x, y\rangle \tag{7}\\
&= \langle x + 0, y + 0\rangle \tag{8}\\
&= \langle x, y\rangle + \mathbf{0} \tag{9}
\end{align}$$

$$\begin{align}
\langle x, y\rangle + \langle x', y'\rangle &= \langle x + x', y + y'\rangle \tag{10}\\
&= \langle x' + x, y' + y\rangle \tag{11}\\
&= \langle x', y'\rangle + \langle x, y\rangle \tag{12}
\end{align}$$

- $(\mathbb{R} \times \mathbb{R}, \otimes, \mathbf{1})$ must be a monoid with identity element $\mathbf{1}$:

$$\begin{align}
(\langle x, y\rangle \otimes \langle x', y'\rangle) \otimes \langle x'', y''\rangle &= \langle x \cdot x', x \cdot y' + y \cdot x'\rangle \otimes \langle x'', y''\rangle \tag{13}\\
&= \langle x \cdot x' \cdot x'', x \cdot x' \cdot y'' + (x \cdot y' + y \cdot x') \cdot x''\rangle \tag{14}\\
&= \langle x \cdot x' \cdot x'', x \cdot x' \cdot y'' + x \cdot y' \cdot x'' + y \cdot x' \cdot x''\rangle \tag{15}\\
&= \langle x, y\rangle \otimes \langle x' \cdot x'', x' \cdot y'' + y' \cdot x''\rangle \tag{16}\\
&= \langle x, y\rangle \otimes (\langle x', y'\rangle \otimes \langle x'', y''\rangle) \tag{17}
\end{align}$$

$$\begin{align}
\mathbf{1} \otimes \langle x, y\rangle &= \langle 1, 0\rangle \otimes \langle x, y\rangle \tag{18}\\
&= \langle 1 \cdot x, 1 \cdot y\rangle \tag{19}\\
&= \langle x, y\rangle \tag{20}\\
&= \langle x \cdot 1, y \cdot 1\rangle \tag{21}\\
&= \langle x, y\rangle \otimes \mathbf{1} \tag{22}
\end{align}$$

- Multiplication left and right distributes over addition:

$$\begin{align}
\langle x, y\rangle \otimes (\langle x', y'\rangle \oplus \langle x'', y''\rangle) &= \langle x, y\rangle \otimes \langle x' + x'', y' + y''\rangle \tag{23}\\
&= \langle x \cdot x' + x \cdot x'', x \cdot y' + x \cdot y'' + y \cdot x' + y \cdot x''\rangle \tag{24}\\
&= \langle x \cdot x', x \cdot y' + y \cdot x'\rangle \oplus \langle x \cdot x'', x \cdot y'' + y \cdot x''\rangle \tag{25}\\
&= (\langle x, y\rangle \otimes \langle x', y'\rangle) \oplus (\langle x, y\rangle \otimes \langle x'', y''\rangle) \tag{26}
\end{align}$$

$$\begin{align}
(\langle x, y\rangle \oplus \langle x', y'\rangle) \otimes \langle x'', y''\rangle &= \langle x + x', y + y'\rangle \otimes \langle x'', y''\rangle \tag{27}\\
&= \langle x \cdot x'' + x' \cdot x'', x \cdot y'' + x' \cdot y'' + y \cdot x'' + y' \cdot x''\rangle \tag{28}\\
&= \langle x \cdot x'', x \cdot y'' + y \cdot x''\rangle \oplus \langle x' \cdot x'', x' \cdot y'' + y' \cdot x''\rangle \tag{29}\\
&= (\langle x, y\rangle \otimes \langle x'', y''\rangle) \oplus (\langle x', y'\rangle \otimes \langle x'', y''\rangle) \tag{30}
\end{align}$$

- Multiplication by $\mathbf{0}$ annihilates $\mathbb{R} \times \mathbb{R}$:

$$\begin{align}
\mathbf{0} \otimes \langle x, y\rangle &= \langle 0, 0\rangle \otimes \langle x, y\rangle \tag{31}\\
&= \langle 0 \cdot x, 0 \cdot y\rangle \tag{32}\\
&= \langle 0, 0\rangle \tag{33}\\
&= \mathbf{0} \tag{34}\\
&= \langle 0, 0\rangle \tag{35}\\
&= \langle x \cdot 0, y \cdot 0\rangle \tag{36}\\
&= \langle x, y\rangle \otimes \langle 0, 0\rangle \tag{37}\\
&= \langle x, y\rangle \otimes \mathbf{0} \tag{38}
\end{align}$$

b) Our initial graph looks like Fig. 1.

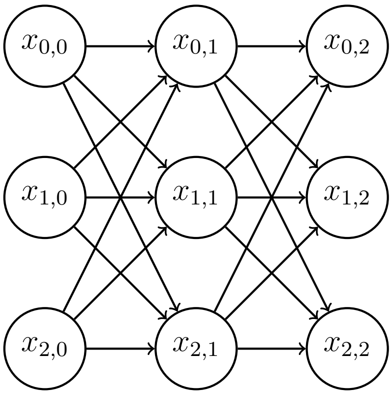

Figure 1: The initial graph

Where the columns represent the words in $\mathbf{w}$ and the rows represent different tags. We use the algorithm from the script:

Algorithm 1: Forward pass

```
$\beta(\mathbf{w}, t_0) = 1$
for $n = 1 \to N$ do
    $\beta(\mathbf{w}, t_n) = \sum_{t_{n-1} \in \mathcal{T}} \exp(\text{score}_\theta(\langle\langle t_{n-1}, t_n\rangle\rangle, \boldsymbol{w})) \otimes \beta(\mathbf{w}, t_{n-1})$
end
```

When we now lift the CRF into the expectation semiring, the forward propagation algorithm changes to:

Algorithm 2: Forward pass

```
$\beta(\mathbf{w}, t_0) = \langle 1, 0 \rangle$
**for** $n = 1 \rightarrow N$ **do**
    $\beta(\mathbf{w}, t_n) = \oplus_{t_{n-1} \in \mathcal{T}} \langle w, -w \log w \rangle \otimes \beta(\mathbf{w}, t_{n-1})$
**end**
```

Where $w = \exp(\text{score}_\theta(\langle\langle t_n, t_{n+1} \rangle\rangle, \boldsymbol{w}))$.
The output of the forward algorithm lifted into the semiring will yield:

$$\bigoplus_{t_{1:N} \in T^n} \bigotimes_{n=1}^{N} \langle w, -w \log w \rangle \tag{39}$$

We want to show that the result of the forward propagation lifted in the semiring is the same as the unnormalized Entropy:

$$H_u(T_w) = -\sum_{\mathbf{t} \in \mathcal{T}^N} \exp(\mathit{score}_{\boldsymbol{\theta}}(\mathbf{t}, \boldsymbol{w}))\mathit{score}_{\boldsymbol{\theta}}(\mathbf{t}, \boldsymbol{w}) \tag{40}$$

$$\tag{41}$$

We show this by induction. Starting with the base case where $N = 1$:

$$\begin{aligned}
\bigoplus_{t_1 \in T^1} \bigotimes_{n=1}^{1} \langle w, -w \log w \rangle = \bigoplus_{t_1 \in T^1} & \langle \exp(\text{score}_\theta(\langle t_0, t_1 \rangle, \boldsymbol{w})), \qquad \qquad \qquad (42) \\
& - \exp(\text{score}_\theta(\langle t_0, t_1 \rangle, \boldsymbol{w})) \log(\exp(\text{score}_\theta(\langle t_0, t_1 \rangle, \boldsymbol{w}))) \rangle \\
& \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad (43) \\
= \bigoplus_{t \in T} & \Big\langle \exp(\text{score}_\theta(t, \boldsymbol{w})), \\
& - \exp(\text{score}_\theta(t, \boldsymbol{w})) \log(\exp(\text{score}_\theta(t, \boldsymbol{w}))) \Big\rangle \qquad (44) \\
= & \Big\langle \sum_{t \in T} \exp(\text{score}_\theta(t, \boldsymbol{w})), \\
& - \sum_{t \in T} \exp(\text{score}_\theta(t, \boldsymbol{w})) \log(\exp(\text{score}_\theta(t, \boldsymbol{w}))) \Big\rangle \qquad (45)
\end{aligned}$$

Our induction hypothesis is the following:

$$\bigoplus_{t_{1:i} \in T^i} \bigotimes_{n=1}^{i} \langle w, -w \log w \rangle = \Big\langle \sum_{t \in T^i} \exp(\text{score}_\theta(t, \boldsymbol{w})), -\sum_{t \in T^i} \exp(\text{score}_\theta(t, \boldsymbol{w})) \log(\exp(\text{score}_\theta(t, \boldsymbol{w}))) \Big\rangle \tag{46}$$

Meaning we assume that $\beta(\mathbf{w}, t_i)$ corresponds to the unnormalized entropy of all sequences of lenght i.

Now we proceed with the induction step, where $i \to i+1$:

$$\begin{aligned}
&\bigoplus_{t_{1:i+1}\in T^{i+1}} \bigotimes_{n=1}^{i+1} \langle w, -w \log w\rangle \\
&= \bigoplus_{t_{1:i+1}\in T} \left(\bigoplus_{t_{1:i}\in T^{i}} \bigotimes_{n=1}^{i} \langle w, -w \log w\rangle \right) \otimes \langle w, -w \log(w)\rangle \qquad (47) \\
&\overset{\text{I.H.}}{=} \bigoplus_{t_{1:i+1}\in T} \left\langle \sum_{t\in T^i} \exp(\text{score}_\theta(t, \boldsymbol{w})), -\sum_{t\in T^i} \exp(\text{score}_\theta((,\boldsymbol{w})t))\text{score}_\theta(t,\boldsymbol{w}) \right\rangle \\
&\otimes \langle \exp(\text{score}_\theta(\langle t_i, t_{i+1}\rangle, \boldsymbol{w})), -\exp(\text{score}_\theta(\langle t_i, t_{i+1}\rangle, \boldsymbol{w}))\text{score}_\theta(\langle t_i, t_{i+1}\rangle, \boldsymbol{w})\rangle \qquad (48)
\end{aligned}$$

We will now analyze both parts of the semiring separately:

$$\begin{aligned}
&\left(\sum_{t\in T^i} \exp(\text{score}_\theta(t, \boldsymbol{w})) \right) \exp(\text{score}_\theta(\langle t_i, t_{i+1}\rangle, \boldsymbol{w})) \qquad (49) \\
&= \sum_{t\in T^i} \exp(\text{score}_\theta(t, \boldsymbol{w})) + \text{score}_\theta(\langle t_i, t_{i+1}\rangle, \boldsymbol{w}) \qquad (50)
\end{aligned}$$

And the second part:

$$\begin{aligned}
&\left(\sum_{t\in T^i} \exp(\text{score}_\theta(t, \boldsymbol{w})) \right) - \exp(\text{score}_\theta(\langle t_i, t_{i+1}\rangle, \boldsymbol{w}))\text{score}_\theta(\langle t_i, t_{i+1}\rangle, \boldsymbol{w}) \\
&\quad + \left(-\sum_{t\in T^i} \exp(\text{score}_\theta((,\boldsymbol{w})t))\text{score}_\theta(t, \boldsymbol{w}) \right) \exp(\text{score}_\theta(\langle t_i, t_{i+1}\rangle, \boldsymbol{w})) \qquad (51) \\
&= -\sum_{t\in T^i} \exp(\text{score}_\theta(t, \boldsymbol{w}) + \text{score}_\theta(\langle t_i, t_{i+1}\rangle, \boldsymbol{w}))\text{score}_\theta(\langle t_i, t_{i+1}\rangle, \boldsymbol{w}) \\
&\quad - \left(\sum_{t\in T^i} \exp(\text{score}_\theta((,\boldsymbol{w})t))\text{score}_\theta(t, \boldsymbol{w}) \right) \exp(\text{score}_\theta(\langle t_i, t_{i+1}\rangle, \boldsymbol{w})) \qquad (52) \\
&= -\sum_{t\in T^i} \exp(\text{score}_\theta(t, \boldsymbol{w}) + \text{score}_\theta(\langle t_i, t_{i+1}\rangle, \boldsymbol{w}))\text{score}_\theta(\langle t_i, t_{i+1}\rangle, \boldsymbol{w}) \\
&\quad - \sum_{t\in T^i} \exp(\text{score}_\theta(t, \boldsymbol{w}) + \text{score}_\theta(\langle t_i, t_{i+1}\rangle, \boldsymbol{w}))\text{score}_\theta(t, \boldsymbol{w}) \qquad (53) \\
&= -\sum_{t\in T^i} \exp(\text{score}_\theta(t, \boldsymbol{w}) + \text{score}_\theta(\langle t_i, t_{i+1}\rangle, \boldsymbol{w}))\left(\text{score}_\theta(t, \boldsymbol{w}) + \text{score}_\theta(\langle t_i, t_{i+1}\rangle, \boldsymbol{w})\right) \\
&\qquad (54)
\end{aligned}$$

We can now combine the two parts to get with eq. (48):

$$\bigoplus_{t_{1:i+1}\in T} \left\langle \sum_{t\in T^i} \exp(\text{score}_\theta(t,\boldsymbol{w})), -\sum_{t\in T^i} \exp(\text{score}_\theta((,\boldsymbol{w})t))\text{score}_\theta(t,\boldsymbol{w}) \right\rangle$$
$$\otimes \langle \exp(\text{score}_\theta(\langle t_i, t_{i+1}\rangle, \boldsymbol{w})), -\exp(\text{score}_\theta(\langle t_i, t_{i+1}\rangle, \boldsymbol{w}))\text{score}_\theta(\langle t_i, t_{i+1}\rangle, \boldsymbol{w})\rangle \tag{55}$$

$$\overset{\text{def. }\otimes}{=} \bigoplus_{t_{1:i+1}\in T} \left\langle \left(\sum_{t\in T^i} \exp(\text{score}_\theta(t,\boldsymbol{w}))\right) \exp(\text{score}_\theta(\langle t_i, t_{i+1}\rangle, \boldsymbol{w})),\right.$$
$$\left(\sum_{t\in T^i} \exp(\text{score}_\theta(t,\boldsymbol{w}))\right) - \exp(\text{score}_\theta(\langle t_i, t_{i+1}\rangle, \boldsymbol{w}))\text{score}_\theta(\langle t_i, t_{i+1}\rangle, \boldsymbol{w})$$
$$\left. + \left(-\sum_{t\in T^i} \exp(\text{score}_\theta((,\boldsymbol{w})t))\text{score}_\theta(t,\boldsymbol{w})\right) \exp(\text{score}_\theta(\langle t_i, t_{i+1}\rangle, \boldsymbol{w}))\right\rangle \tag{56}$$

$$\overset{eq.\ (50)}{=} \bigoplus_{t_{1:i+1}\in T} \left\langle \sum_{t\in T^i} \exp(\text{score}_\theta(t,\boldsymbol{w})) + \text{score}_\theta(\langle t_i, t_{i+1}\rangle, \boldsymbol{w}),\right.$$
$$\left(\sum_{t\in T^i} \exp(\text{score}_\theta(t,\boldsymbol{w}))\right) - \exp(\text{score}_\theta(\langle t_i, t_{i+1}\rangle, \boldsymbol{w}))\text{score}_\theta(\langle t_i, t_{i+1}\rangle, \boldsymbol{w})$$
$$\left. + \left(-\sum_{t\in T^i} \exp(\text{score}_\theta((,\boldsymbol{w})t))\text{score}_\theta(t,\boldsymbol{w})\right) \exp(\text{score}_\theta(\langle t_i, t_{i+1}\rangle, \boldsymbol{w}))\right\rangle \tag{57}$$

$$\overset{eq.\ (54)}{=} \bigoplus_{t_{1:i+1}\in T} \left\langle \sum_{t\in T^i} \exp(\text{score}_\theta(t,\boldsymbol{w})) + \text{score}_\theta(\langle t_i, t_{i+1}\rangle, \boldsymbol{w}),\right.$$
$$\left. -\sum_{t\in T^i} \exp(\text{score}_\theta(t,\boldsymbol{w}) + \text{score}_\theta(\langle t_i, t_{i+1}\rangle, \boldsymbol{w}))\,(\text{score}_\theta(t,\boldsymbol{w}) + \text{score}_\theta(\langle t_i, t_{i+1}\rangle, \boldsymbol{w}))\right\rangle \tag{58}$$

$$\overset{\text{def. }\oplus}{=} \left\langle \sum_{t_{i+1}\in T}\sum_{t\in T^i} \exp(\text{score}_\theta(t,\boldsymbol{w})) + \text{score}_\theta(\langle t_i, t_{i+1}\rangle, \boldsymbol{w}),\right.$$
$$\left. \sum_{t_{i+1}\in T}\sum_{t\in T^i} \exp(\text{score}_\theta(t,\boldsymbol{w}) + \text{score}_\theta(\langle t_i, t_{i+1}\rangle, \boldsymbol{w}))\,(\text{score}_\theta(t,\boldsymbol{w}) + \text{score}_\theta(\langle t_i, t_{i+1}\rangle, \boldsymbol{w}))\right\rangle \tag{59}$$

$$= \left\langle \sum_{t\in T^{i+1}} \exp(\text{score}_\theta(t,\boldsymbol{w})), -\sum_{t\in T^{i+1}} \exp(\text{score}_\theta(t,\boldsymbol{w}))\text{score}_\theta(t,\boldsymbol{w}) \right\rangle \tag{60}$$

Which concludes our induction proof and we have shown that eq. (40) holds.

c) We want to prove:

$$H(T_w) = Z(\boldsymbol{w})^{-1}H_U(T) + \log(Z(\boldsymbol{w}) \tag{61}$$

$$H(T_w) = -\sum_{t \in T^N} p(t \mid w) \cdot \log(p(t \mid w)) \tag{def. H}$$

(62)

$$= -\sum_{t \in T^N} \frac{\exp(\text{score}_\theta(t, \boldsymbol{w}))}{Z(\boldsymbol{w})} \log\left(\frac{\exp(\text{score}_\theta(t, \boldsymbol{w}))}{\sum_{t' \in T^N} \exp(\text{score}_\theta(t', \boldsymbol{w}))}\right) \tag{def. p}$$

(63)

$$= -\sum_{t \in T^N} \frac{\exp(\text{score}_\theta(t, \boldsymbol{w}))}{Z(\boldsymbol{w})} \log\left(\frac{\exp(\text{score}_\theta(t, \boldsymbol{w}))}{Z(\boldsymbol{w})}\right) \tag{64}$$

$$= -\sum_{t \in T^N} \frac{\exp(\text{score}_\theta(t, \boldsymbol{w}))}{Z(\boldsymbol{w})} (\text{score}_\theta(t, \boldsymbol{w}) - \log Z(\boldsymbol{w})) \tag{65}$$

$$= -\sum_{t \in T^N} \frac{\exp(\text{score}_\theta(t, \boldsymbol{w}))\text{score}_\theta(t, \boldsymbol{w}) - \exp(\text{score}_\theta(t, \boldsymbol{w})) \log Z(\boldsymbol{w}))}{Z(\boldsymbol{w})} \tag{66}$$

$$= -\sum_{t \in T^N} \frac{\exp(\text{score}_\theta(t, \boldsymbol{w}))\text{score}_\theta(t, \boldsymbol{w})}{Z(\boldsymbol{w})} + \sum_{t \in T^N} \frac{\exp(\text{score}_\theta(t, \boldsymbol{w})) \log Z(\boldsymbol{w}))}{Z(\boldsymbol{w})}$$

(67)

$$= H_U(T_{\boldsymbol{w}}) Z(\boldsymbol{w})^{-1} + \frac{\log(Z(\boldsymbol{w}))}{Z(\boldsymbol{w})} \sum_{t \in T^N} \exp(\text{score}_\theta(t, \boldsymbol{w})) \tag{68}$$

$$= H_U(T_{\boldsymbol{w}}) Z(\boldsymbol{w})^{-1} + \log(Z(\boldsymbol{w})) \tag{69}$$

(70)

d) We want to show that $H(T_w)$ can be computed in $\mathcal{O}(N \cdot |\mathcal{T}|^2)$.
We do this by looking at the identity given in the previous subquestion and show that each term can be computed in at most $\mathcal{O}(N \cdot |\mathcal{T}|^2)$. As stated in the exercise $\log(Z(w))$ can be computed in $\mathcal{O}(N \cdot |\mathcal{T}|^2)$. In exercise b) we have shown that with the expectation semiring we can calculate $H_U(T_w)$ with one forward/backward pass. Algorithm 2 shows that the forward pass can be done in $\mathcal{O}(N \cdot |\mathcal{T}|)$, for the outer loop over the word length and for each iteration the sum over all possible taggings. The last term, $Z(w)$, can also be calculated in $\mathcal{O}(N \cdot |\mathcal{T}|)$, by running a forward/backwards pass without a semiring. This is shown in the lectures "Efficiently Computing the Normalizer", where we use the distributive property of the product to calculate the normalizer with a linear number of terms. When we have all the subterms we only need a multiplication and an addition to arrive at $H(T_w)$.

We can calculate the gradient by doing backpropagation over its computation graph. Hence, this can be done in the same bound as $\mathcal{O}(N \cdot |T|^2)$

Question 2:

a) We again show the correctness by induction.

Base case ($|w| = 1$)**:**
Our base case is a sequence with length 1. After dequeueing the first element from the queue, which is the initialization element, we push elements for all possible taggings of

the first word. In the next iteration, due to the priority queue structure we dequeue the element with the highest score. This is now a complete tagging for our sequence of length 1 and it has the highest score.

Induction hypothesis:
The first sequence of length i that is popped from the queue has the highest score among all sequences of length i.

Induction step $(i \rightarrow i+1)$:
Here we have two cases, one where the best tagging for sequence of length $i+1$ contains the best tagging for sequence of length i as a prefix and one where it does not.
Case 1 $(t_{1:i} \in t_{1:i+1})$:
Here $t_{1:i}$ denotes the a tagging for $w_1, \dots, w_i$. This case is straight forward, as per our induction hypothesis we know that $t_{1:i}$ is the best tagging for sequence of length i and will therefore be the first element for sequence of length i that is popped from the queue. The reasoning is the same as in the base case, all possible taggings of the next word are pushed to the queue and the element with the highest score is popped next.
Case 2 $(t_{1:i} \notin t_{1:i+1})$:
Because our scores are in $\mathcal{R}_{\leq 0}$ we know that when we add to a sequence we can only decrease the score. Assume that the best tagging for length $i+1$ $T_{1:i+1}$ has score s_{i+1}. We then have two cases for all taggings $t_{1:k}$ where $k \leq i$. Either they are smaller or equal to s_{i+1}, then we do not care about them, since they cannot be used for the best tagging because we can only decrease them even more. Or they are larger than s_{i+1}, then we know that they will get popped before $T_{1:i+1}$, because they have a higher score. Therefore we know that the subtagging of $T_{1:i+1}$ $t'_{1:i}$ will be popped before any tagging with length $i+1$. This holds by contradiction, as otherwise $T_{1:i+1}$ would not be the best tagging.

b) We want to show that as Viterbi calculates the best tagging of lenght i and ending with tag t in $\gamma[i,t]$, so does Dijkstra.
First we look at the dimension of γ, we know that for the Viterbi algorithm γ has dimensions $\mathcal{R}^{N \otimes |\mathcal{T}|}$. We can clearly see that Dijkstra matches these dimensions, as for every possible tuple of tags $t \in \mathcal{T}$ and lenght $1 \leq i \leq N$ we have a score in γ. This holds because we save all pairs in an array and only allow distinct pairs in our queue (priority is updated when we have an overlap).
To prove that the values are equivalent as well, we perform a contradiction proof:
We assume that the score s_i of tagging $T_{1:i}$ is not the best tagging of length i and it's score is saved in $\gamma[i,t]$. Then there must exist a tagging $T'_{1:i}$ with score s'_i that is better than $T_{1:i}$. Going back to the algorithm and our argument of exercise a) we know that all subtaggings $T'_{1:k}$ for $k \leq i$ have a higher score than s'_i and therefore must have been popped from the queue before $\langle i, t \rangle$ and hence $\gamma[i,t]$ would be set to s'_i. This contradicts our assumption and therefore s_i must be the best tagging of length i and it's score is saved in $\gamma[i,t]$.

c) The normal Dijkstra with a priority queue based on a heap runs in $\mathcal{O}(|V|\log(|V|)+|E|)$. Our graph has $|\mathcal{T}| \cdot |\mathbf{w}|$ vertices as we have all possible taggins for each word. The edges are bound by $|\mathcal{T}|^2 \cdot |\mathbf{w}|$. This bound holds, becasuse each tag for a word has a connection to all other tags for the next word. When combining these insights we get the following runtime bound:

$$\mathcal{O}(|\mathcal{T}||\mathbf{w}|\log(|\mathcal{T}||\mathbf{w}|) + |\mathcal{T}|^2 \cdot |\mathbf{w}|) \tag{71}$$

Viterbi:

$$\mathcal{O}(|\mathbf{w}| \cdot |T|^2) \tag{72}$$

d) The problem with the given semiring, is that multiple scores for a given tagging can affect the priority. This is a problem because we want the priority to be representative of the best score and not depend on other scores, which may be much worse. Hence, this semiring would not always have the best tagging of a given lenght i popped first.

e) The property that we need from our semiring, are the ones that we use in our proof. One of which is $s_{1:i} \geq s_{1:i+1}$. Since we are using the $\otimes$ operator to build $s_{1:i+1}$ from $s_{1:i}$, this gives as constraints on the operator. Our priority queue must always return the tagging with the best score, hence we need the $\oplus$ to be a max operator. We could also have used a min operator, but then we would have to negate the scores.

Question 3:

Link to Colab Notebook: `https://colab.research.google.com/drive/1lFGpgB3MTTJZGb9akxGsRegdf`
`p?usp=sharing`
Question e):
The naive viterbi implementation is clearly the slowest, which matches our expectation. Dijkstra is the second fastest, with a runtime of 34.8 ms and backward viterbi is the fastest with 25.1 ms (local runtimes, but matches results on colab). These observations match our expectations. I did the training for and g) outside of the notebook, just with a single python script as I was getting reduced performance and errors when running in the notebook.
Output for g):

```
Starting training with beta 1.0
Epoch 0, batch 0
Epoch 0, batch 20
...
Epoch 0, batch 360
Epoch 0, batch 380
-------------------------
Epoch: 1 / 3
Development set accuracy: 0.8933365941047668
-------------------------
Epoch 1, batch 0
Epoch 1, batch 20
...
Epoch 2, batch 360
Epoch 1, batch 380
-------------------------
```

```
Epoch: 2 / 3
Development set accuracy: 0.906506359577179
-------------------------
Epoch 2, batch 0
Epoch 2, batch 20
...
Epoch 2, batch 360
Epoch 2, batch 380
-------------------------
Epoch: 3 / 3
Development set accuracy: 0.9176903367042542
-------------------------

Starting training with beta 10.0
Epoch 0, batch 0
Epoch 0, batch 20
...
Epoch 0, batch 360
Epoch 0, batch 380
-------------------------
Epoch: 1 / 3
Development set accuracy: 0.8931728601455688
-------------------------
Epoch 1, batch 0
Epoch 1, batch 20
...
Epoch 1, batch 360
Epoch 1, batch 380
-------------------------
Epoch: 2 / 3
Development set accuracy: 0.9124546051025391
-------------------------
Epoch 2, batch 0
Epoch 2, batch 20
...
Epoch 2, batch 360
Epoch 2, batch 380
-------------------------
Epoch: 3 / 3
Development set accuracy: 0.9162563681602478
-------------------------

Starting training with beta 0.1
Epoch 0, batch 0
Epoch 0, batch 20
...
Epoch 0, batch 360
Epoch 0, batch 380
-------------------------
Epoch: 1 / 3
Development set accuracy: 0.9024679660797119
-------------------------
Epoch 1, batch 0
Epoch 1, batch 20
...
Epoch 1, batch 360
Epoch 1, batch 380
-------------------------
Epoch: 2 / 3
Development set accuracy: 0.9182560443878174
-------------------------
Epoch 2, batch 0
Epoch 2, batch 20
...
Epoch 2, batch 360
Epoch 2, batch 380
-------------------------
Epoch: 3 / 3
Development set accuracy: 0.9221615791320801
-------------------------
```