NLP assignment #1

Andrius Buinovskij - 18-940-270

Q1 b) i)

Well since all inputs are 1 and all weights are 1 then

$$\mathbf{s}_1^1 = \sum_{i=1}^{3} \mathbf{x}_i \cdot \mathbf{w}_{i,1}^1 = 3 \tag{1}$$

$$\mathbf{s}_2^1 = \sum_{i=1}^{3} \mathbf{x}_i \cdot \mathbf{w}_{i,2}^1 = 3 \tag{2}$$

Where $\mathbf{s}_j^i$ is the sum input to the j'th neuron in the i'th layer, and $\mathbf{s}^i$ is a vector whose length is equal to the number of neurons in the i'th layer, with the input being the 0'th layer. So $\mathbf{s}^1$ is of length 2 since 1'st layer has 2 neurons.

Let $\mathbf{n}_j^i$ be the output of the j'th neuron in the i'th layer, then of course $\mathbf{n}^i$ is a vector of length equal to the number of neurons in the i'th layer:

$$n_j^i = ReLU\big(\mathbf{s}_j^i\big) \tag{3}$$

So in our case We get

$$\mathbf{n}_1^1 = ReLU\big(\mathbf{s}_1^1\big) = ReLU\big(3\big) = 3 \tag{4}$$

$$\mathbf{n}_2^1 = ReLU\big(\mathbf{s}_2^1\big) = ReLU\big(3\big) = 3 \tag{5}$$

Same steps in the next layer:

$$\mathbf{s}_1^2 = 3 \cdot 1 + 3 \cdot 1 = 6 \tag{6}$$

And now We pass this through a sigmoid for our output instead of a *ReLU* so We get

$$out = \sigma\big(\mathbf{s}_1^2)\big) = \frac{1}{1 + e^{-6}} = 0.99752737684 \tag{7}$$

Q1 b) ii)

$$\frac{d}{d\mathbf{w}^1_{j,k}}\mathbf{x}_i \cdot \mathbf{w}^1_{j,k} = \mathbf{x}_i \tag{8}$$

Since the inputs are all 1 this simplifies to

$$\frac{d}{d\mathbf{w}^1_{j,k}}\mathbf{x}_i \cdot \mathbf{w}^1_{j,k} = \frac{d}{d\mathbf{w}^1_{j,k}}1 \cdot \mathbf{w}^1_{j,k} = 1 \tag{9}$$

Now for the second layer:

$$\frac{d}{d\mathbf{w}^2_{1,1}}f = \frac{d}{d\mathbf{w}^2_{1,1}}\mathbf{n}_{1,1} \cdot \mathbf{w}^2_{1,1} = \frac{d}{d\mathbf{w}^2_{1,1}}ReLU\big(\mathbf{s}_{1,1}\big) \cdot \mathbf{w}^2_{1,1} = ReLU\big(\mathbf{s}_{1,1}\big) \tag{10}$$

Since $ReLU\big(\mathbf{s}_{1,1}\big)$ is just a constant w.r.t. $\mathbf{w}^2_{1,1}$. In this case We simply get

$$\frac{d}{d\mathbf{w}^2_{1,1}}f = ReLU\big(\mathbf{s}_{1,1}\big) = 3 \tag{11}$$

And likewise for the other weight.

Q1 b) iii)

$$\begin{aligned} L_{BCE} &= -(y\log(\hat{y}) + (1-y)\log(1-\hat{y})) && (12)\\ &= -(0 \cdot \log(0.99752737684) + (1-0)\log(1-0.99752737684)) && (13)\\ &= -(\log(0.00247262316)) && (14)\\ &= 2.60684206722 && (15) \end{aligned}$$

redone

Alright let's just roll all of these questions into one and do an iteration of backprop.

The forward pass is trivial.

Let $\mathbf{x}^l_i$ be the output of the i'th neuron in the l'th layer.

Let $\mathbf{w}^l_{i,j}$ be the weight belonging to j'th neuron multiplying the i'th input in the l'th layer. $\mathbf{w}^l_{\cdot,j}$ is then simply the weight vector associated with the j'th neuron in the l'th layer.

Let $\mathbf{s}^l_j$ be $(\mathbf{x}^{l-1})^\top \mathbf{w}^l_{\cdot,j}$, the weighted sum of inputs to the j'th neuron in the l'th layer.

We can then say that $x^l_i = \sigma(\mathbf{s}^l_j)$, where σ is the nonlinearity of choice.

Then define

$$\delta^l_j = \frac{\partial L_{BCE}}{\partial \mathbf{s}^l_j} \tag{16}$$

And finally

$$\begin{aligned}
\delta_j^{l-1} &= \sum_{i=1}^{d^l} \frac{\partial L_{BCE}}{\partial \mathbf{s}_i^l} \frac{\partial \mathbf{s}_i^l}{\partial \mathbf{x}_j^{l-1}} \frac{\partial \mathbf{x}_j^{l-1}}{\partial \mathbf{s}_j^{l-1}} \tag{17} \\
&= \sum_{i=1}^{d^l} \delta_i^l \frac{\partial \mathbf{s}_i^l}{\partial \mathbf{x}_j^{l-1}} \frac{\partial \mathbf{x}_j^{l-1}}{\partial \mathbf{s}_j^{l-1}} \tag{18}
\end{aligned}$$

Where d^l is the number of neurons in layer l. So now We have a recursive definition which uses dynamic programming. This could be further nuanced by expressing things in matrix notation, but it's good enough for present purposes.

Here is also a picture since just looking at symbols is a nightmare.

So now

$$\frac{\partial L_{BCE}}{\partial \mathbf{s}_1^2} = \frac{\partial L_{BCE}}{\partial \hat{y}} \frac{\partial \hat{y}}{\mathbf{s}_1^2} \tag{19}$$

$$\begin{aligned}
\frac{\partial L_{BCE}}{\partial \hat{y}} &= \frac{\partial}{\partial \hat{y}} - \big(y \log(\hat{y}) + (1-y)\log(1-\hat{y})\big) \tag{20} \\
&= -y\frac{1}{\hat{y}} - (1-y)\frac{\partial}{\partial \hat{y}} \log(1-\hat{y}) \tag{21} \\
&= -y\frac{1}{\hat{y}} - (1-y)\frac{-1}{1-\hat{y}} \tag{22} \\
&= -y\frac{1}{\hat{y}} + \frac{1-y}{1-\hat{y}} \tag{23}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \hat{y}}{\mathbf{s}_1^2} &= \frac{\partial}{\mathbf{s}_1^2}\sigma(\mathbf{s}_1^2) \tag{24} \\
&= \sigma(\mathbf{s}_1^2) \cdot (1 - \sigma(\mathbf{s}_1^2)) \tag{25}
\end{aligned}$$

So then We have

$$\begin{aligned}
\delta_1^2 = \frac{\partial L_{BCE}}{\partial \mathbf{s}_1^2} &= \frac{\partial L_{BCE}}{\partial \hat{y}} \frac{\partial \hat{y}}{\mathbf{s}_1^2} \tag{26} \\
&= \left(-y\frac{1}{\hat{y}} + \frac{1-y}{1-\hat{y}} \right)\left(\sigma(\mathbf{s}_1^2) \cdot (1 - \sigma(\mathbf{s}_1^2)) \right) \tag{27}
\end{aligned}$$

Alright. We only have one more of these to figure out:

$$\delta_j^1 = \sum_{i=1}^{d^2} \delta_i^2 \frac{\partial \mathbf{s}_i^2}{\partial \mathbf{x}_j^1} \frac{\partial \mathbf{x}_j^1}{\partial \mathbf{s}_j^1} \tag{28}$$

But since $d^2 = 1$, i.e. there is only one neuron in the output layer, We have

$$\delta_j^1 = \sum_{i=1}^{d^2} \delta_i^2 \frac{\partial \mathbf{s}_i^2}{\partial \mathbf{x}_j^1} \frac{\partial \mathbf{x}_j^1}{\partial \mathbf{s}_j^1} \tag{29}$$

$$= \delta_1^2 \frac{\partial \mathbf{s}_1^2}{\partial \mathbf{x}_j^1} \frac{\partial \mathbf{x}_j^1}{\partial \mathbf{s}_j^1} \tag{30}$$

$$= \delta_1^2 \mathbf{w}_j^2 \sigma(\mathbf{s}_j^1)(1 - \sigma(\mathbf{s}_j^1)) \tag{31}$$

Now for the gradient update We will of course need

$$\frac{\partial L_{BCE}}{\partial \mathbf{w}_{i,j}^l} \tag{32}$$

But this can be easily derived since

$$\frac{\partial L_{BCE}}{\partial \mathbf{w}_{i,j}^l} = \frac{\partial L_{BCE}}{\partial \mathbf{s}_j^l} \frac{\partial \mathbf{s}_j^l}{\partial \mathbf{w}_{i,j}^l} \tag{33}$$

$$= \delta_j^l \mathbf{x}_i^{l-1} \tag{34}$$

Now We do a forward pass and We know that $\mathbf{s}_i^1 = 3$ and $\mathbf{s}_1^2 = 6$. Plugging in values We then get

$$\delta_1^2 = \Bigg(-y\frac{1}{\hat{y}} + \frac{1-y}{1-\hat{y}} \Bigg) \Bigg(\sigma(\mathbf{s}_1^2) \cdot (1 - \sigma(\mathbf{s}_1^2)) \Bigg) \tag{35}$$

$$= \Bigg(\frac{1}{1 - 0.99752737684} \Bigg) \Bigg(\sigma(6) \cdot (1 - \sigma(6)) \Bigg) \tag{36}$$

$$= \Bigg(\frac{1}{1 - 0.99752737684} \Bigg) \Bigg(0.99752737684 \cdot (1 - 0.99752737684) \Bigg) \tag{37}$$

$$= 404.428792942 \cdot 0.00246650929 \tag{38}$$

$$= 0.99752737493 \tag{39}$$

Similarly We have

$$\delta_j^1 = \delta_1^2 \mathbf{w}_j^2 \sigma(\mathbf{s}_j^1)(1 - \sigma(\mathbf{s}_j^1)) \tag{40}$$

$$\delta_1^1 = \delta_1^2 \mathbf{w}_1^2 \sigma(\mathbf{s}_1^1)(1 - \sigma(\mathbf{s}_1^1)) \tag{41}$$

$$= 0.997527374931 \cdot \sigma(3)(1 - \sigma(3)) \tag{42}$$

$$= 0.997527374931 \cdot 0.95257412682 \cdot 0.04742587317 \tag{43}$$

$$= 0.04506495478 \tag{44}$$

δ^1_2 is equal to δ^1_1.
Now that We have the deltas We can use

$$\frac{\partial L_{BCE}}{\partial \mathbf{w}^l_{i,j}} = \delta^l_j \mathbf{x}^{l-1}_i \tag{45}$$

For weights in the first layer this means

$$\begin{align} \frac{\partial L_{BCE}}{\partial \mathbf{w}^1_{i,j}} &= \delta^1_j \mathbf{x}^0_i \tag{46} \\ &= 0.04506495478 \cdot 1 \tag{47} \\ &= 0.04506495478 \tag{48} \end{align}$$

Where the 0'th layer is the input layer.

$$\begin{align} \frac{\partial L_{BCE}}{\partial \mathbf{w}^2_{i,j}} &= \delta^2_j \mathbf{x}^1_i \tag{49} \\ &= 0.99752737493 \cdot 3 \tag{50} \\ &= 2.99258212479 \tag{51} \end{align}$$

All that is left is to perform a step for each weight. All weights in the first layer simplify to

$$\begin{align} w^1_{i,j} &= 1 - 0.1 \cdot 0.04506495478 \tag{52} \\ &= 0.99549350452 \tag{53} \end{align}$$

Sidenote
So when actually implementing this, for the next assignemnt, We get

$$\begin{align} \delta^2_1 = \frac{\partial L_{BCE}}{\partial \mathbf{s}^2_1} &= \frac{\partial L_{BCE}}{\partial \hat{y}} \frac{\partial \hat{y}}{\mathbf{s}^2_1} \tag{54} \\ &= \left(-y\frac{1}{\hat{y}} + \frac{1-y}{1-\hat{y}} \right) \left(\sigma(\mathbf{s}^2_1) \cdot (1 - \sigma(\mathbf{s}^2_1)) \right) \tag{55} \\ &= \left(-y\frac{1}{\hat{y}} + \frac{1-y}{1-\hat{y}} \right) \left(\hat{y} \cdot (1 - \hat{y}) \right) \tag{56} \\ &= -\hat{y} \cdot (1-\hat{y}) \cdot \frac{y}{\hat{y}} + \hat{y} \cdot (1-\hat{y}) \cdot \frac{1-y}{1-\hat{y}} \tag{57} \\ &= -(1-\hat{y}) \cdot y + \hat{y} \cdot (1-y) \tag{58} \end{align}$$

Q4 a) i)

We're going to use linearity of expectation.

Let $I_i, i \in \{1 \dots |V|\}$ be the indicator variable for whether or not the word i appears in Mary's sampling.

Since draws are with replacement and the distribution is uniform, for a token *not* to appear, all n draws must have chosen another token.

The probability of not choosing token i in a particular draw is of course $(|V|-1)/|V|$. The probability of not choosing token i over n samples is then

$$\left(\frac{|V|-1}{|V|}\right)^n \tag{59}$$

However, We are interested in the opposite event - We want to know the probability of *not* not choosing token i, which is given by

$$1-\left(\frac{|V|-1}{|V|}\right)^n \tag{60}$$

By linearity of expectation and the fact that the expected value of an indicator is simply the probability of it being 1, We get

$$E\left(\sum_{i=1}^{|V|} = I_i\right) = \sum_{i=1}^{|V|} E(I_i) = \sum_{i=1}^{|V|} P(I_i = 1) = |V| \cdot \left(1 - \left(\frac{|V|-1}{|V|}\right)^n\right) \tag{61}$$

Q4 a) ii)

Given n draws, what is the probability that all the words will appear in the sample?

All of my attempts have failed, and a whole lot of searching leads me to conclude that the probability is:

$$\frac{|V|!}{|V|^n} S_2(n-1, |V|-1) \tag{62}$$

Where $S_2(n-1, |V|-1)$ is the Stirling number of the second kind.

Q4 b) i)

Alright, so, let X be the number of samples before "work" and "hard" bigram appears.

Let ps (partial success) mean that the word "work" has been sampled. Then by law of total expectation We can say

$$E(X) = (1 + E(X|ps)) \cdot P(ps) + (1 + E(X|ps')) \cdot P(ps') \tag{63}$$

Where We the 1's in there come from the fact that a single sample has been taken.

Now, given partial success We either fully succeed (fs) or fail and go back to the beginning, so:

$$\begin{aligned} E(X|ps) &= (1 + E(X|ps, fs)) \cdot P(fs|ps) + (1 + E(X|ps, fs')) \cdot P(fs'|ps) && (64) \\ &= (1) \cdot \frac{1}{|V|} + (1 + E(X)) \cdot \frac{|V| - 1}{|V|} && (65) \end{aligned}$$

Where given full success, expected number of draws to get a success is zero. Given a failure We're back to $E(X)$.

Of course, given ps', the compliment of partial success i.e. partial failure, $E(X|ps') = E(X)$.

Putting it all together We get:

$$\begin{aligned} E(X) &= (1 + E(X|ps)) \cdot P(ps) + (1 + E(X|ps')) \cdot P(ps') && (66) \\ &= \left(1 + \frac{1}{|V|} + (1 + E(X)) \cdot \frac{|V| - 1}{|V|}\right) \cdot \frac{1}{|V|} + (1 + E(X)) \cdot \frac{|V| - 1}{|V|} && (67) \\ &= \frac{1}{|V|} + \frac{1}{|V|^2} + (1 + E(X)) \cdot \frac{|V| - 1}{|V|^2} + (1 + E(X)) \cdot \frac{|V| - 1}{|V|} && (68) \\ &= \frac{1}{|V|} + \frac{1}{|V|^2} + \frac{|V| - 1}{|V|^2} + \frac{(|V| - 1)E(X)}{|V|^2} + \frac{|V| - 1}{|V|} + \frac{E(X)(|V| - 1)}{|V|} && (69) \end{aligned}$$

Now We just need to simplify this mess.

$$\begin{aligned} E(X) &= \frac{1}{|V|} + \frac{1}{|V|^2} + \frac{|V| - 1}{|V|^2} + \frac{(|V| - 1)E(X)}{|V|^2} + \frac{|V| - 1}{|V|} + \frac{E(X)(|V| - 1)}{|V|} && (70) \\ E(X) - \frac{(|V| - 1)E(X)}{|V|^2} - \frac{E(X)(|V| - 1)}{|V|} &= \frac{1}{|V|} + \frac{1}{|V|^2} + \frac{|V| - 1}{|V|^2} + \frac{|V| - 1}{|V|} && (71) \\ E(X)\left(1 - \frac{(|V| - 1)}{|V|^2} - \frac{(|V| - 1)}{|V|}\right) &= \frac{1}{|V|} + \frac{1}{|V|^2} + \frac{|V| - 1}{|V|^2} + \frac{|V| - 1}{|V|} && (72) \\ E(X) &= \frac{\frac{1}{|V|} + \frac{1}{|V|^2} + \frac{|V|-1}{|V|^2} + \frac{|V|-1}{|V|}}{1 - \frac{(|V|-1)}{|V|^2} - \frac{(|V|-1)}{|V|}} && (73) \end{aligned}$$

Beautiful. If You plug $|V| = 2$ in there You get 6, which is the well known result of expected number of tosses to get 2 heads in a row.

Q4 b) ii)

Well, the probability of "work" not appearing in n draws is

$$\left(\frac{|V| - 1}{|V|}\right)^n \qquad (74)$$

And of course the probability of it appearing is

$$1 - \left(\frac{|V|-1}{|V|}\right)^n \tag{75}$$

And Mary wants this probability to be above 0.95:

$$\begin{align}
1 - \left(\frac{|V|-1}{|V|}\right)^n &\geq 0.95 \tag{76}\\
-\left(\frac{|V|-1}{|V|}\right)^n &\geq -0.05 \tag{77}\\
\left(\frac{|V|-1}{|V|}\right)^n &\leq 0.05 \tag{78}\\
\log\left(\left(\frac{|V|-1}{|V|}\right)^n\right) &\leq \log(0.05) \tag{79}\\
n \cdot \log\left(\frac{|V|-1}{|V|}\right) &\leq \log(0.05) \tag{80}\\
n &\geq \frac{\log(0.05)}{\log\left(\frac{|V|-1}{|V|}\right)} \tag{81}\\
\tag{82}
\end{align}$$

Where the last sign flips since log of a quantity less than 1 is negative.

Q4 c) i)

The answer here is identical to the answer to question **b)i)**, by the fact that the tokens are all equally likely.

Q4 c) ii)

Okay so We have

$$h_t^0 = f(w_0 x_{t-1} + w_1 x_t + w_2 h_{t-1}^0 + b_0) \tag{83}$$

So, that's our first hidden state. f is some arbitrary non-linearity (or maybe linearity, whatever, a function).

w_0 multiplies previous input, w_1 multiplies current input, w_2 multiplies the previous hidden state and We have a bias term.

The output then is

$$y_t = g(w_3 h_t^0 + b_1) \tag{84}$$

So w_3 is multiplying the hidden state, there's a bias term and another function.

The goal is to output x_{t-1} and x_t are the same word.

Okay so assume that the two words are equal and the bias is 0:

$$
\begin{aligned}
h_t^0 &= f(w_0 x_{t-1} + w_1 x_t + w_2 h_{t-1}^0 + b_0) && (85)\\
&= f(w_0 x_{t-1} + w_1 x_{t-1}) && (86)\\
&= f(x_{t-1}(w_0 + w_1)) && (87)
\end{aligned}
$$

So clearly one way to go is to let w_0 be -1 and w_1 be 1, so that when they are added, We get zero in the case of equality. Then f can just be like a check as to whether the input is 0, and if it is output 1, otherwise output 0.

g is then simply the identity function and b_1 is also 0.

So, $w_0 = -1$, $w_1 = 1$, $w_2 = 0$, $w_3 = 1$, all biases are zero, f is a boolean check for whether the input is 0 and g is the identity function.

Q4 c) iii)

So We simply use the second hidden layer as a counter. $w_4, w_5 = 1, b_1, b_2 = 0$ and g and h are still the identity function, and f is the boolean 0 check.

Q4 c) iv)

I think the non-uniform unigram language model has the greater probability of drawing two of the same token in a row.

This isn't a proof, but the reasoning is as follows: let p_i be the probability of drawing the i'th token. Now suppose We alter the distribution such that $p_i' = p_i - x$. Now, We'll have to add that x to some other p_j to preserve a distribution, and since the sample are identical, sampling p_j twice in a row will be more likely than if We had just left the probabilities alone.

Q5 a)
Well first let's re-derive this mess.

$$\sum_{\mathbf{t}\in\mathcal{T}^N} \exp\left\{\sum_{n=1}^{N} \text{score}\big(\langle t_{n-1}, t_n\rangle, \mathbf{w}\big)\right\} \tag{88}$$

So let's start with inside the bracket comes from

$$\text{score}\big(\mathbf{t}, \mathbf{w}\big) = \sum_{n=1}^{N} \text{score}\big(\langle t_{n-1}, t_n\rangle, \mathbf{w}\big) \tag{89}$$

So this is our simplifying assumption, and I suppose it's also what makes this a "Conditional Random Field". **t** stands for "tag" and it's the, well, tagging of the word sequence **w**. So We're saying that the score for the tag does not have to be calculated all in one go, but rather can be done in parts, where each part only depends on it's predecessor. Aight. Markov yay.

N is the length of the sentence by the way, so the length of both **t** and **w**.

$\mathcal{T}$ is the set of all possible tags for any particular word. Since there are N words, the total number of possible tags for **w** is then $\mathcal{T}^N$.

So then the expression is simply calculating the normalizing constant, since it's summing over all possible tags $\mathbf{t} \in \mathcal{T}^N$.

So, up next We get

$$\sum_{\mathbf{t}\in\mathcal{T}^N} \exp\left\{\sum_{n=1}^{N} \text{score}\big(\langle t_{n-1}, t_n\rangle, \mathbf{w}\big)\right\} \tag{90}$$

$$= \sum_{\mathbf{t}_{1:n}\in\mathcal{T}^N} \prod_{n=1}^{N} \exp\left\{\text{score}(\langle \mathbf{t}_{n-1}, \mathbf{t}_n\rangle, \mathbf{w})\right\} \tag{91}$$

Okay so what have We done here. For one, **t** now has a subscript. Sure.

We've converted the inner sum to a product, which is fine - before We were taking the exponent of a sum, but product of a bunch of exponents will sum the exponents so no worries there. We then still sum over every possible tag. Sure.

$$\sum_{\mathbf{t}\in\mathcal{T}^N} \exp\left\{\sum_{n=1}^{N} \text{score}\big(\langle t_{n-1}, t_n\rangle, \mathbf{w}\big)\right\} \tag{92}$$

$$= \sum_{\mathbf{t}_{1:n}\in\mathcal{T}^N} \prod_{n=1}^{N} \exp\left\{\text{score}(\langle \mathbf{t}_{n-1}, \mathbf{t}_n\rangle, \mathbf{w})\right\} \tag{93}$$

$$= \sum_{\mathbf{t}_{1:N-1}\in\mathcal{T}^{N-1}} \sum_{t_N\in\mathcal{T}} \prod_{n=1}^{N} \exp\left\{\text{score}(\langle \mathbf{t}_{n-1}, \mathbf{t}_n\rangle, \mathbf{w})\right\} \tag{94}$$

So, the first change is that the subscript for $\mathbf{t}$ now goes to $N-1$. So We're just taking out the last tag in the sequence of tags $\mathbf{t}_N$.

Since that last tag could be anything, We sum over all the possible tags, so $t_N \in \mathcal{T}$. There are $|\mathcal{T}|$ choices.

I mean really You could rewrite the first sum as N sums each over $|\mathcal{T}|$ terms each. So We're just splitting one out. No worries.

$$\sum_{\mathbf{t}\in\mathcal{T}^N} \exp\left\{\sum_{n=1}^{N} \mathrm{score}\big(\langle t_{n-1}, t_n\rangle, \mathbf{w}\big)\right\} \tag{95}$$

$$= \sum_{\mathbf{t}_{1:n}\in\mathcal{T}^N} \prod_{n=1}^{N} \exp\left\{\mathrm{score}(\langle \mathbf{t}_{n-1}, \mathbf{t}_n\rangle, \mathbf{w})\right\} \tag{96}$$

$$= \sum_{\mathbf{t}_{1:N-1}\in\mathcal{T}^{N-1}} \sum_{t_N\in\mathcal{T}} \prod_{n=1}^{N} \exp\left\{\mathrm{score}(\langle \mathbf{t}_{n-1}, \mathbf{t}_n\rangle, \mathbf{w})\right\} \tag{97}$$

$$= \sum_{\mathbf{t}_{1:N-1}\in\mathcal{T}^{N-1}} \prod_{n=1}^{N-1} \exp\left\{\mathrm{score}(\langle \mathbf{t}_{n-1}, \mathbf{t}_n\rangle, \mathbf{w})\right\} \times \sum_{t_N\in\mathcal{T}} \exp\left\{\mathrm{score}(\langle \mathbf{t}_{N-1}, \mathbf{t}_N\rangle, \mathbf{w})\right\} \tag{98}$$