CMPT 413/825: Natural Language Processing **(Due: 10/29/20)**

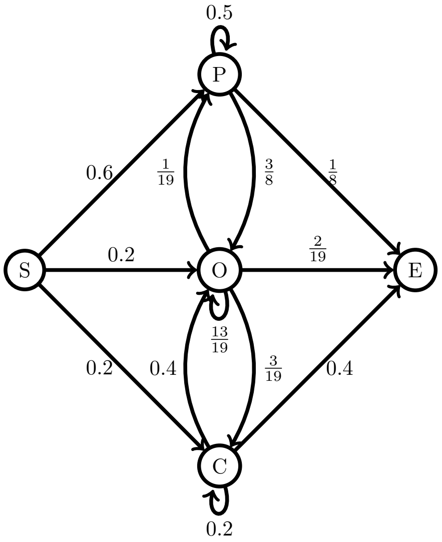
Homework 3, Fall 2020

Name: Coulton Fraser
SFU Email: coultonf@sfu.ca, *ID:* 301405411 *Students discussed with:*

Course Policy: Read all instructions carefully before you start working on the assignment, and before you make a submission. The course assignment policy is available at https://angelxuanchang.github.io/nlp-class/.

Problem 1. Hidden Markov Models

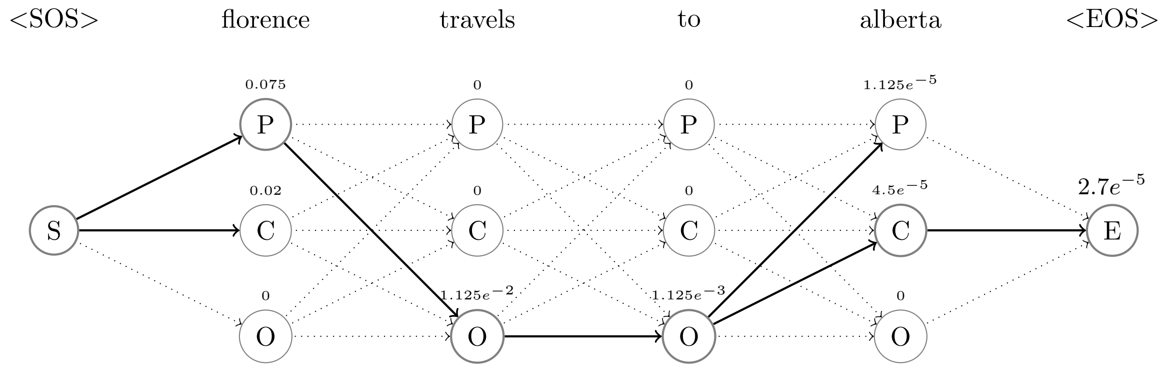
(a)



(b)

	florence	washington	madison	new	york	robert	nick	denzel
city (c)	0.5	0.5	0.5	1	1	0	1	1
person (p)	0.5	0.5	0	0	0	1	0	0

(c)



Most likely annotation: SPOOCE

Problem 2. LSTM Gradients

(a)

$$\begin{aligned}
\overline{h^t} &= \overline{i^{(t+1)}}w_{(ih)} + \overline{f^{(t+1)}}w_{(fh)} + \overline{o^{(t+1)}}w_{(oh)} \\
\overline{c^{(t)}} &= \overline{h^{(t)}}(1 - \tanh^2(c^{(t)})) \\
\overline{g^{(t)}} &= \overline{c^{(t)}}i^{(t)} \\
\overline{o^{(t)}} &= \overline{h^{(t)}}\tanh(c^{(t)}) \\
\overline{f^{(t)}} &= \overline{c^{(t)}}c^{(t-1)} \\
\overline{i^{(t)}} &= \overline{c^{(t)}}g^{(t)}
\end{aligned} \tag{0.1}$$

(b)

$$\overline{w_{(ix)}} = \overline{i^{(t)}}x^{(t)} \tag{0.2}$$

(c) LSTM cells have a forget gate which enables the gradient to eb controlled at each step so that it doesn't vanish. As well, the addition of the gates inside the LSTM cell help to disperse the gradient to four weights (i,o,f,g) rather than just one making it more balanced.

Problem 3. Neural Sequence Models

(a) The hidden state and the context as a vector, and a starting input word or previous words that were predicted from the decoder.

(b) The context vector allows a more likely translation as words are decoded according to the context being built, and the previous words, act as a means to determine the current decoded context that has been produced. Therefore, these affect inference by at each time step the next most likely word given the previous context will be chosen according to the previous context generated from the encoder.

(c) One strategy is the greedy method, where the most likely previously decoded words are fed into the next time step of the decoder. The downside of this is that the best translation may be lost, but a benefit is that it is very fast.
Another strategy is the beam method, where the k most likely previously decoded words are fed into the next time step of the decoder and then the resulting k-tree is reversed through to find the most optimal combination of words. The benefit of this is that it is more likely to find the optimal translation without sacrificing too much computing to find the optimal translation. A downside is that it is more expensive than the greedy method.

(d) GRU's are more efficient than LSTM due to their simplicity compared to the more complex LSTM. The GRU would be more ideal if there are limited resources for training as it has half the parameters to learn, and also if the LSTM tends to overfit the data. If there is a small amount of data, the GRU could possibly perform better as it is less likely to overfit.

(e) Bidirectional GRU:
Because we have access to the entire input sequence when doing classification we can use bidirectional GRU's that has two hidden states. One for the default forward pass and an additional one for the backward pass. The bidirectional GRU will increase performance as it will not rely on the left-right sequence nearly as much as it would without a reverse hidden state included. This may help with text classifiers as a classification doesn't depend as much on order as something like sequence prediction. Therefore, the bidirectional allows the sentence to be represented better as a collection of words contributing to a classifier context.

Multilayer GRU:
Adding stacked GRU's to the model will allow more hidden states to be created, thus capturing more features from the input text. The output of the hidden state for a time step is fed into the next layer of some depth for the same time step as input. This might help as a classification requires the text as a whole to be understood. Adding extra depth to a GRU could allow the model to build context of words in a deep layer in the multilayer network. At a shallower depth GRU stacked on the previous layer GRU, the model could learn to represent the other hidden state in a more abstract concept to learn the classifier better. This helps because it allows the model to capture many hidden states each throughout the text.

Problem 4. BLEU Score

(a)

$$\begin{aligned} c_1, p_1 &= 0.6 \\ c_1, p_2 &= 0.5 \\ c_2, p_1 &= 0.8 \\ c_2, p_2 &= 0.5 \\ \\ len(c) &= 4 \\ len(r) &= 5 \\ \\ BP &= e^{(\frac{1}{5})} \\ \\ BLEU_{c1} &= 0.669... \\ BLEU_{c2} &= 0.772... \end{aligned} \tag{0.3}$$

C2 is considered the better BLEU translation. I think that this is accurate.

(b)

$$\begin{aligned} c_1, p_1 &= 0.6 \\ c_1, p_2 &= 0.5 \\ c_2, p_1 &= 0.4 \\ c_2, p_2 &= 0.25 \\ \\ len(c) &= 6 \\ len(r) &= 5 \\ \\ BP &= 1 \\ \\ BLEU_{c1} &= 0.5477... \\ BLEU_{c2} &= 0.3162... \end{aligned} \tag{0.4}$$

C1 is considered the better BLEU translation. I think that C2 is the better translation.

(c) As we can see in our previous examples, if we only have one reference translation, then the BLEU score can be improved through words that aren't relevent to the meaning of the sentence by chance. The meaning of the sentence vs the composition of the sentence is different, but if you only give BLEU one reference, then the composition can potentially have a larger impact.

(d) Advantages:
- Its very easy to use and is automatic.
- Its been shown to have a positive correlation with actual translations, so can be considered accurate in that regard.

Disadvantages:
- If there are just a lot of similar words between two sentences it will give a high score regardless of if it is correct.
- In order for it to be accurate, large n-grams 4 or higher seem to be necessary, leading to it not working with shorter sentences.