CMPT 413/825: Natural Language Processing **(Due: 11/12/20)**

Homework 4, 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. Analyzing NMT Errors

a) Error: Repeated favorite word.
Reasoning: The model may lose attention to specific words.
Solution: Implement an additive attention model.

b) Error: Second half of sentence produces words in the wrong order.
Reasoning: Possibly due to a loss of context further in the sentence.
Solution: Increase hidden layers as the sentence loses its ability to translate the longer the translation becomes.

c) Error: Unknown name, Bolingbroke, not recognized
Reasoning: The model might not be given an method for dealing with unknown words.
Solution: Implement a copy mechanism that will copy out-of-vocabulary words.

d) Error: Manzana is not being translated correctly
Reasoning: Manzana has two translations, apple and block.
Solution: Could try to implement a Luong attention mechanism. The idea is that by focusing on parts of the sentence it will be clear that an apple is not part of the context of the sentence.

e) Error: Translating profesores as women.
Reasoning: Likely because in the beginning of the sentence, there is lots of context around the word - 'She' and teacher's lounge being a feminine word in Spanish indicated by la.
Solution: Bidirectional encoder-decoder model would allow the end of sentence to have equal weight as the beginning, which would reduce the impact of She on the context of teacher's lounge.

f) Error: Unit conversion - hectares - acres
Reasoning: Both are a unit of measurement, and therefore similarity may be found between the two which is incorrect.
Solution: Implementing an attention model would allow more focus on each word, allowing many connections for different types of measurement. Doing additive attention mechanism may be sufficient.

Problem 2. Contextualized Word Embeddings

(a) An advantage of character level embedding is that the model will be able to handle out-of-vocabulary word better as many words which may have been unseen before can be some form, root of, or combination of previously seen words allowing the model to better understand unknown words.

(b) Masked language modelling is the process where the network will mask out, replace a word with another, or keep the word as is, and try to predict the replaced word during the training phase. The objective is to be able to predict the masked word given the context. The advantage of using this is that the model is forced to learn the context of words, rather than just the typical relationships between words.

(c) During training the span of entity, and the sentence as word pieces would be provided as input because BERT can take learn the relations between two text inputs. The output of the BERT model would be a predicted span entity for DBpedia. The parameters to be learned would be the attention matrix. The training objective is to learn the relationship between the two input sentences.

(d) The text is treated completely and un-sequentially, that is given a specific word in a sentence, it is compared to every other word regardless of position. Another reason transformers outperform RNN's is that they have a self-attention layer that allows the network to associate words together in a sentence as a means of understanding word connectivity.

Problem 3. Constituency Parsing

(a)

$$
\begin{aligned}
N &= \{S, NP, VP, V, Ajd, C, D, P, Pro\} \\
\Sigma &= \left\{ \begin{array}{cccccccccc} I & ordered & fried & chicken & and & coke & it & rains & this & time \\ of & the & year & little & kids & are & playing & violin & at & concert \end{array} \right\} \\
S &= \{S\}
\end{aligned}
\tag{0.1}
$$

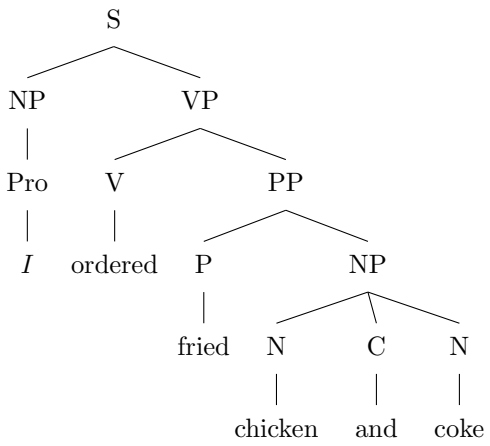
(b)

S	→	NP VP
NP	→	Pro
NP	→	Adj N C N
VP	→	V NP
VP	→	V VP
Pro	→	I
Pro	→	it
D	→	the
D	→	this

20 non-terminals to terminals,
10 non-terminals to non-terminals.

(c) The parameter $q = (\alpha_i \leftarrow \beta_i)$ must be calculated for each rule, which is the probability of each rule according to its parent.

(d)



(e) They can bring better phrasing representations such as embeddings for words tags and nodes. There are learned scoring functions.

Problem 4. Transition-Based Dependency Parsing

(a) The algorithm is linear time in respect to the number of words because there are n shifts plus n arcs. O(2n) = O(n)

(b)

	stack	buffer	action	added arc
0	[ROOT]	[He, drove, to, the, store, to, buy, gifts]	SHIFT	
1	[ROOT, He]	[drove, to, the, store, to, buy, gifts]	SHIFT	
2	[ROOT, He, drove]	[to, the, store, to, buy, gifts]	LEFT-ARC(r)	(drove, r, He)
3	[ROOT, drove]	[to, the, store, to, buy, gifts]	SHIFT	
5	[ROOT, drove, to]	[the store, to, buy, gifts]	SHIFT	
6	[ROOT, drove, to, the]	[store, to, buy, gifts]	SHIFT	
7	[ROOT, drove, to, the, store]	[to, buy, gifts]	LEFT-ARC(r)	(store, r, the)
8	[ROOT, drove, to, store]	[to, buy, gifts]	RIGHT-ARC(r)	(to, r, store)
9	[ROOT, drove, to]	[to, buy, gifts]	RIGHT-ARC(r)	(drove, r, to)
10	[ROOT, drove]	[to, buy, gifts]	SHIFT	
11	[ROOT, drove, to]	[buy, gifts]	SHIFT	
12	[ROOT, drove, to, buy]	[gifts]	SHIFT	
13	[ROOT, drove, to, buy, gifts]	[]	RIGHT-ARC(r)	(buy, r, gifts)
14	[ROOT, drove, to, buy]	[]	RIGHT-ARC(r)	(to, r, buy)
15	[ROOT, drove, to]	[]	RIGHT-ARC(r)	(drove, r, to)
16	[ROOT, drove]	[]	RIGHT-ARC(r)	(ROOT, r, drove)
17	[ROOT]	[]	DONE	

(c) Getting the training data for the dependency parser can be expensive as it requires a linguist to label sentences. Lots of data sparsity in the dependencies due to word-to-word interactions. The dependencies are greatly affected by this during training.

(d) The neural network would be predicting a softmax of the most likely dependency parse, and its inputs would be a vector of words, arc tags, and pos tags.