Wei Han

CS 224n: Assignment #5

February 11, 2020

This assignment is split into two sections: *Neural Machine Translation with RNNs* and *Analyzing NMT Systems*. The first is primarily coding and implementation focused, whereas the second entirely consists of written, analysis questions. If you get stuck on the first section, you can always work on the second. That being said, the NMT system is more complicated than the neural networks we have previously constructed within this class and takes about **4 hours to train on a GPU**. Thus, we strongly recommend you get started early with this assignment. Finally, the notation and implementation of the NMT system is a bit tricky, so if you ever get stuck along the way, please come to Office Hours so that the TAs can support you.

1 Character-based convolutional encoder for NMT (36 points)

(a)

The characteristic of characters are simpler than words, thus its embeddings is lower.

(b)

1. Word LSTM:
$$n_{parameters} = V_word \times k = 25000$$
2. Char LSTM:
$$n_{parameters} = V_char \times k = 480$$
Word-based embedding model has more characters.

(c)(2 points)

(d)(4 points)

(e)(f)(g)(h)(j)(k)

(l)

We get the final BLEU of 99.2979.

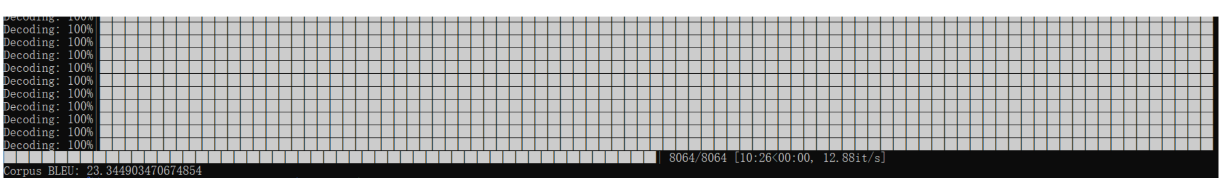
2 Character-based LSTM decoder for NMT (26 points)

(a)(b)(c)(d) (17 points)

(e) (3 points)

(f) (6 points)

Finally we get a BLEU of 23.34.



3 Analyzing NMT Systems (8 points)

(a) (2 points)

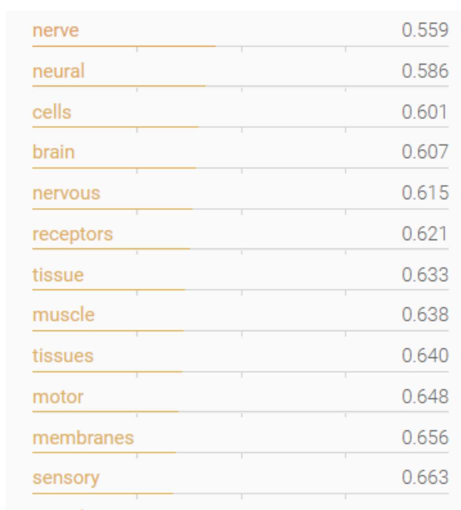
Tranducia 43517; Tranduce 8764, Others didn't occur.
If we use word-level NMT model, then some form of the word cannot be intepreted out because they are not in the vocabulary, and we can only get a <unk> token. But if we use character-level decoder then this problem can be solved. As character-level decoder only capture the relevance between consequential characters hence it can discover some suffix or prefix in other words and so that it's able to produce unrecorded forms of the word.

(b) (2 points)

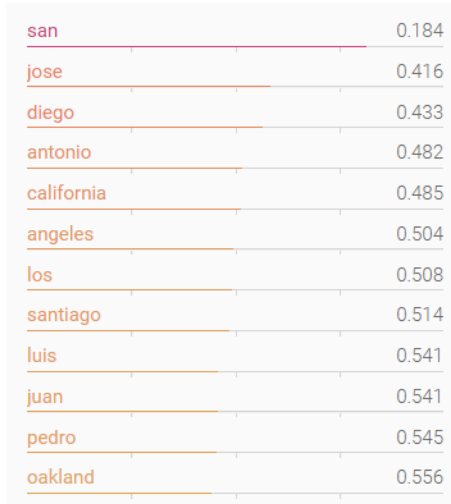
i. (0.5 points)

- financial

- neuron



- Francisco

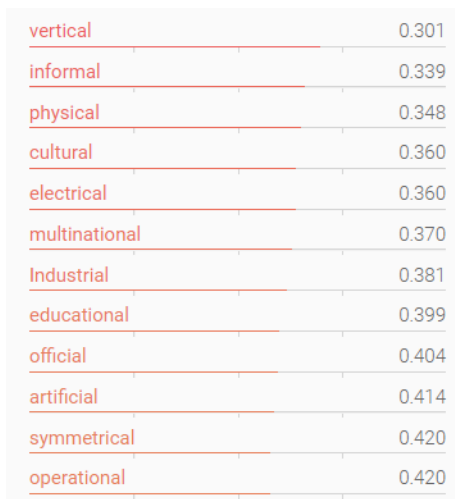


- naturally

- expectation

ii. (0.5 points)

- financial



- neuron

Newton	0.354
George	0.383
NBA	0.404
Delhi	0.415
golden	0.421
person	0.421
Google	0.427
Virgin	0.428
folk	0.430
garden	0.440
monkeys	0.447
Florida	0.450

- Francisco

France	0.420
platform	0.436
tissue	0.451
Foundation	0.459
microphone	0.460
issue	0.492
friend	0.498
charity	0.498
grandfather	0.508
calcium	0.511
mission	0.513
punishment	0.513

- naturally

practically	0.302
typically	0.353
significantly	0.372
mentally	0.375
gradually	0.388
physically	0.400
socially	0.413
particularly	0.419
locally	0.428
generally	0.432
Especially	0.435
safely	0.436

- expectation

iii. (3 points)
Word2Vec captures the meaning similarity of words and word-level sequence dependency.
CharCNN captures the spelling similarity of words.

(c) (2 points)

1. Puedo vestirme como agricultor, o con ropa de cuero, y nunca nadie ha elegido un agricultor.
2. I can dress as a farmer, or in leather clothes, and no one has ever chosen a farmer
3. I can <unk> like farmer, or with leather <unk> and no one has chosen a farmer.
4. I can dress as a farmer, or with leather clothes, and never one has chosen a farmer.
5. It's an acceptable example. It can generate words that never appeared in training data by capturing characteristics of word construction.

1. Estoy desilusionada que de adultos nunca llegamos a conocernos
2. I am disappointed that as adults we never get to know each other
3. I'm <unk> that we have never come to know us.
4. I'm despite adults, we never get to meet us.
5. It's an acceptable example. It wrongly produces the correct words.