Home Assignment 4

1. A window into NER

(a)

i. Examples of sentences containing a named entity with an ambiguous type:

- "Yesterday I visited Castro." Castro may be either ORG or PER.
- "Jordan is beautiful." Jordan may be either LOC or PER.

ii. It might be important to use features apart from the word itself to predict named entity labels because the word might be rare. Using features (that are common to a group of words) helps generalize from the training data to other data.

iii. Features that would help in predicting whether a word is part of a named entity or not:

- Word neighbors and their labels
- Capitalized first letter
- POS tag

(b)

i. e(t), W and U dimensions for a window of size w:

$$e^{(t)}: \ 1 \times D(2w+1)$$
$$W \ : \ D(2w+1) \times H$$
$$U \ : \ H \times C$$

ii. Computational complexity of predicting labels for a sentence of length T:

$$O\left(T\left(HD\left(2w+1\right)+HC\right)\right)$$

Calculation is according to the number of multiplications, as the additions and embeddings operations are not from a higher degree.

(d)

i. Best development entity-level F1 score: 0.84

Corresponding token-level confusion matrix:

go\gu	**PER**	**ORG**	**LOC**	**MISC**	**O**
PER	2945.00	40.00	51.00	19.00	94.00
ORG	128.00	1645.00	117.00	70.00	132.00
LOC	45.00	93.00	1880.00	34.00	42.00
MISC	39.00	57.00	40.00	1027.00	105.00
O	33.00	52.00	14.00	30.00	42630.00

Normalized matrix:

go\gu	PER	ORG	LOC	MISC	O
PER	93.52	1.27	1.62	0.60	2.99
ORG	6.12	78.63	5.59	3.35	6.31
LOC	2.15	4.44	89.78	1.62	2.01
MISC	3.08	4.50	3.15	80.99	8.28
O	0.08	0.12	0.03	0.07	99.70

It can be seen from the confusion matrix that:

- The model identifies well whether a word is a named entity or not.
- The most "problematic" class is ORG.
- The class with the best performance is PER.
- In most cases (3 out of 4) the top ranked class after the gold class is O. In a way, it can be considered as the model acknowledgment of his inability to classify correctly.
- Misclassified LOC tend to be classified as ORG.

ii. Modeling limitations of the window-based model:

- Can't capture dependencies that are longer than the window. For example:

```
x : May 27-29 v Gloucestershire or Sussex or Surrey ( three days )
y*: O   O     O ORG             O  ORG    O  ORG    O O     O    O
y': O   O     O ORG             O  LOC    O  ORG    O O     O    O
```

  In our model the window size is 1, meaning the window for the word "Sussex" is "or Sussex or", which doesn't add information. For a window size 2, the corresponding window is "Gloucestershire or Sussex or Surrey", which gives more information as Sussex is comparable to Gloucestershire and Surrey.

- The model isn't aware of the predicted label of the surrounding words when it does its prediction. Thus, the labels might be inconsistent for one entity. For example:

```
x : IPO FILING -- Homegate Hospitality Inc .
y*: O   O      O  ORG     ORG         ORG O
y': ORG O      O  PER     ORG         ORG O
```

  The word "Homegate" was identified as a named entity, but was miss classified as PER. Knowing that the next 2 labels are ORG might cause the model to classify "Homegate" as ORG as well (one label for the whole entity).

2. RNNs for NER

(a)

i. There are $H^2 - 2wDH = H(H - 2wD)$ more parameters in the RNN model in comparison to the window-based model (for the W_h matrix, for input of 1 word instead of $2w + 1$ words).

ii. Computational complexity of predicting labels for a sentence of length T:

$$O\left(T\left(H^2 + DH + HC\right)\right) = O\left(TH\left(H + D + C\right)\right)$$

Calculation is according to the number of multiplications, as the additions and embeddings operations are not from a higher degree.

(b)

i. A scenario in which decreasing the cross-entropy cost would lead to a decrease in entity-level F1 scores:

An entity that is composed of at least 2 words, and all the words are misclassified as O (not a named entity). Decreasing the CE might cause one of the word to be classified correctly. In that case, the number of detected entities will increase in 1, while the number of correct detected entities will not change. Meaning the precision will decrease: $p = \frac{x}{y} \rightarrow \frac{x}{y+1}$. Since recall is the number of true detected entities divided by the true number of entities, it will not change. Thus, F1 will decrease as well.

ii. It is difficult to directly optimize for F1, because its calculation based on the prediction for all the training data set (impossible to do SGD and divide to batches). In addition, it uses "argmax" (on the softmax output, instead of using it directly) which is not differentiable.

(d)

i. If we didn't use masking, the loss would include the loss of the padded NULL tokens, which are irrelevant. In that way the net will (also) study from irrelevant samples, and it will affect the gradients updates and the parameters values.

Masking solve this problem by omitting the padded zeros from the loss and hence from the gradient updates.

(g)

i. limitations of this RNN model:

- RNN "sees" the past, but not the future while predicting current time step label.

  For example:

  `x : Comelf , based in the central Transylvanian town of Bistrita ,`

```
y*: ORG    O O     O  O   O       MISC          O    O  LOC      O
y': PER    O O     O  O   O       LOC           O    O  LOC      O
```

The word “Comelf” was miss labeled as PER instead of ORG. The information in the rest of the sentence (“ based in the central…”) could help predicting “Comelf” is an organization and not a person.

- The model prediction is done per word and not per named entity. Thus, the labels might be inconsistent for one entity. For example:

```
x : The Federal Republic of  Yugoslavia is the only country of the former
y*: O   LOC     LOC      LOC LOC        O  O   O    O       O  O   O
y': O   ORG     LOC      LOC LOC        O  O   O    O       O  O   O
```

The token “Federal” was identified as a named entity, but was miss classified as ORG. The other 3 tokens of that entity were classified correctly. Thus, prediction for the entire named entity would result in correct classification for the first token as well.

- Can’t capture long dependencies. For example:

```
x : at Driefontein Consolidated and Gold Fields '  Kloof Gold Mining Co this
y*: O  ORG         ORG          ORG ORG  ORG    ORG ORG   ORG  ORG    ORG O
y': O  ORG         ORG          O   ORG  ORG    O   ORG   ORG  ORG    ORG O
```

In this sentence, there is a named entity with 10 tokens. However, the model identified it as 3 separated entities, and classified them correctly. It might stem from the fact the model can’t capture this long dependency.

ii. Model extensions to overcome the limitations:

- Bi-directional RNN “sees” both the past and the future.
- Extend the loss function in a way that it will predict the label of the whole named entity and not each token separately. (For example, model the labels transition probabilities and perform decoding at test time).
- Using LSTM or GRU might help.

3. GRUs and TensorBoard

(c)

i. For a single timestep prediction, the maximum entropy value possible is:

$\ln C = \ln 5 \approx 1.6$

ii. Graphs analysis (graphs are attached separately)

- Average loss

  Reduced as training step increase.

- Logits histogram

  As training step increase, the logits range becomes wider (in both positive and negative directions). That means that the softmax output is getting away from a uniform distribution (similar probabilities for each class), toward probabilities of values close to 0 and 1.

- Average entropy

  Reduced toward 0 as training step increase. Meaning the affective size of the probability space becomes smaller. In other word, as training proceeds, the model considers less classes at each step and becomes more confident in his prediction: the predicted probabilities become closer to 0 and 1. (This is reflected in the increasing range of the logits histogram).

(d)

According to the model output, the best predicted class is O, and after it the class PER.

The model struggles (relatively) on the MISC prediction.

In addition, the model suffers from the same limitations described in the RNN part.

The model performance is not always reflected by the prediction values. In many cases where an O token was labeled correctly as O, the prediction probability is relatively low. On the other hand, it can be seen, that in most cases of incorrect classification, the prediction value is low.

For example:

x:	it	would	fan	sectarian	tensions	in	**British-ruled**	Northern	Ireland	.
y*:	O	O	O	O	O	O	**MISC**	LOC	LOC	O
y':	O	O	O	O	O	O	**LOC**	LOC	LOC	O
p :	1.00	1.00	1.00	1.00	1.00	1.00	**0.95**	0.97	1.00	1.00

The above sentence demonstrates the problem of mixing labels for 1 entity. It also can be seen that the model is confident in the correct predictions, while in the mistaken prediction, the probability is lower.