KU Leuven - NLP 2021-2022
26 Oct 2021

Exercise Session 1

Named Entity Recognition, Relation Extraction, Part-of-speech tagging

For questions contact:
Toledo Forum for Questions or
nlp@ls.kuleuven.be

As a resource for this exercise, you can make use of the book **Jurafsky and Martin - Speech and Language Processing (3rd draft), chapter 8**. We will make use of the Universal Dependencies tagset. A list of this tagset is given in Figure 8.1 of the book.
As an optional practice, there is a coding exercise. Note that this is not graded and will not be discussed at the exam. Only written questions will be part of the exam.

1 Computing the POS Tag Predictions

In this exercise we will practice with the hidden markov model for POS tagging. We will make us of a bi-gram assumption. First we need a set of probabilities. We will make use of the following dataset of sentences.

```
SOS/SOS the/DET fly/NOUN sits/VB at/ADP morning/NOUN time/NOUN
SOS/SOS my/PRON work/NOUN makes/VB the/DET time/NOUN fly/VB
SOS/SOS I/PRON fly/VB in/ADP the/DET morning/NOUN
SOS/SOS I/PRON like/VB breakfast/NOUN in/ADP the/DET morning/NOUN
SOS/SOS I/PRON work/VB before/ADP I/PRON fly/VB
SOS/SOS the/DET fly/NOUN sits/VB in/ADP my/PRON breakfast/NOUN
```

Question 1.A: Define The Markov Chain

To predict the POS tags for new sentences, we will make use of a HMM. Before we can do that, we need to define our Markov Chain and compute all its probabilities.

1. Use the dataset given above to draw the Markov chain. Only draw the chain for the hidden states, the POS tags.
2. With the Markov chain defined, compute all the needed probabilities. There are two probabilities we need for computing:
   - The transition probability: This indicate the probability of going from one state to another, in our case the tags T in a sequence: $P(T_i + 1|T_i)$
   - The state observation probability: this indicates how likely an observation is produced from our state. In our case, the is the probability of a word w being predicted from a tag t: $P(w|t)$

Question 1.B: Assign POS tags

Given the computed probabilities from the above exercise and the defined Markov Chain, make use of the HMM to compute the most probable POS tags for all words in the following sentence. We assume a greedy model.

- I fly before I work

2 How good is our NER?

We have trained two NER models and we would like to test them. However, we forgot to create a test set so we can evaluate their performance. Your task in this exercise is to create the test annotations. Using these annotations we will do an evaluation of the predictions from the models and compare the results.
Our label set L consists of the following items: **Person**, **Role**, and **O** if it is neither.
Here is the sentence and the predictions for our models:

Sentence	Model1	Model2	GT
Barack	O	Person	P
Obama	O	Person	P
was	O	O	O
a	O	O	O
great	O	O	O
President	O	Role	R
with	O	O	O
Michelle	O	O	P
Obama	O	O	P
as	O	Person	O
his	O	O	O
First	O	O	O
Lady	O	Role	R
who	O	Role	O
did	O	O	O
important	O	O	O
work	O	Role	O

Question 2.A: Annotate the sentence

Given the sentence in the table above, assign the correct labels.

Question 2.B: Evaluating the models

Now we have the correct annotations for our test sentence, we can finally see which models performs best. Let's use two metrics: the **accuracy** and the **Macro-F1**. The Macro-F1 metric combines the F1 scores of each label. And the F1 is a metric that balances the precision (P) and recall (R). Use the following definition:

$$MacroF1 = \frac{1}{|L|} \sum_{l \in L} F1_l$$

$$F1 = 2 \cdot \frac{P \cdot R}{P + R}$$

1. For each of the metrics, which is the best performing model?
2. Which metric is the better to use for this task, and why? Explain by making use of the results from subquestion 2.B.1.

3 Simple RNN Encoder-Decoder for RE

In this Exercise we will do some Relation Extraction(RE) using an Encoder-Decoder model. We will make use of an RNN for both the encoder and the decoder. To simplify our models, the decoder will only predict a single relation per sentence, so it will always predict three elements. The first token is the predicate, followed by two entities: `<relation, entity1, entity2>`.
We start by giving all the weight matrices and bias vectors of the encoder-decoder model.

- For the Encoder we have $\boldsymbol{W}_{(in)}$, $\boldsymbol{W}_{(enc,h)}$, and $\boldsymbol{b}_{(enc,h)}$.
- For the Decoder we have $\boldsymbol{W}_{(in)}$, $\boldsymbol{W}_{(dec,h)}$, and $\boldsymbol{b}_{(dec,h)}$.
- For the output we need two different layers:
  - One for predicting predicates with $\boldsymbol{W}_{(out,pred)}$, and $\boldsymbol{b}_{(out,pred)}$
  - One for predicting entities $\boldsymbol{W}_{(out,ent)}$, and $\boldsymbol{b}_{(out,ent)}$.

Note, that that both the encoder and the decoder share the $\boldsymbol{W}^{(in)}$ matrix for encoding the input.

We limit our vocabulary, so it only contains the words we need. Our vocabulary contains the following **ordered** words:

is TA nlp victor isTA PresidentOf SOS

With TA being the abbreviation for Teaching-Assistant and nlp for Natural Language Processing.

The possible predicates are:

isTA PresidentOf

Note that the predicates are also included in the vocabulary.
We use one hot embedding to represent the words in the vocabulary.

Important: for simplicity we don't use any non-linearity in our RE model.

Here are all the needed weights and biases:

$$\boldsymbol{W}^{(in)} = \begin{bmatrix} 0.3 & -0.2 \\ -1.1 & 0.8 \\ 0.5 & 1 \\ 0.4 & -0.5 \\ -0.2 & -0.2 \\ 0.9 & 0.5 \\ 0 & 0 \end{bmatrix}$$

$$\boldsymbol{W}^{(enc,h)} = \begin{bmatrix} 1.3 & -0.5 \\ -0.5 & 0.5 \end{bmatrix} \qquad \boldsymbol{b}^{(enc,h)} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$\boldsymbol{W}^{(dec,h)} = \begin{bmatrix} -1 & 0.1 \\ 0.4 & 0.7 \end{bmatrix} \qquad \boldsymbol{b}^{(dec,h)} = \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}$$

$$\boldsymbol{W}^{(out,pred)} = \begin{bmatrix} 0.6 & -0.2 \\ 0.5 & 0.8 \end{bmatrix} \qquad \boldsymbol{b}^{(out,pred)} = \begin{bmatrix} 0.4 \\ 0.8 \end{bmatrix}$$

$$\boldsymbol{W}^{(out,ent)} = \begin{bmatrix} 0.2 & 0.3 & 0.4 & 0.3 & 0.2 & 0.1 & 0 \\ 1 & -0.2 & 0.5 & 0.8 & -0.8 & 0.4 & 0 \end{bmatrix} \qquad \boldsymbol{b}^{(out,ent)} = \begin{bmatrix} 1 \\ 2 \\ 1 \\ 2 \\ 1 \\ 2 \\ 1 \end{bmatrix}$$

Question 3.A: Drawing the Network

With the definition of the model above, you should be able to draw a representation of the entire model. Try to indicate clearly what the dimensions are of the different layers.

Question 3.B: Predict the Relation

We will now predict the relation for the following sentence using this network. We initialise the hidden state at timestep 0 as $h^{(0)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$. What is the predicted relation for the sentence **"Victor is nlp TA"**? And what is the **probability** of this prediction?

Question 3.C: Some open questions

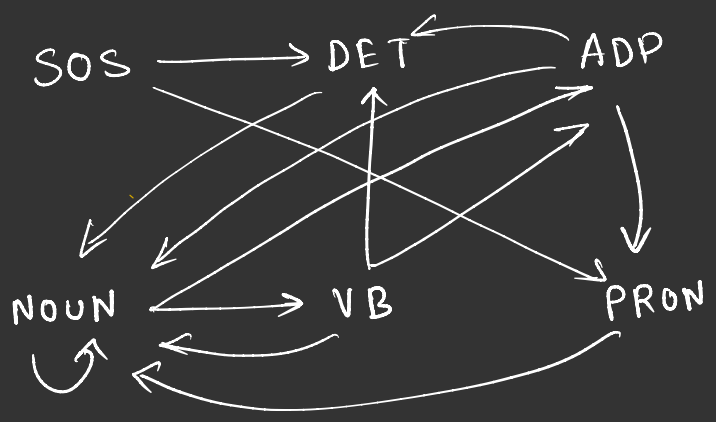
1. Currently, we can only predict a single relation triplet, what do we need to change to predict multiple triplets?
2. Currently, we simply use an RNN Encoder. What is the advantage of changing this to a Bidirectional-RNN Encoder? In that case what needs to be changed in your model?

4 OPTIONAL CODING: Implement a POS Tagger

NOTE: There won't be any programming on the exam and there will be no grading for the coding exercises. For this assignment, we will look into the implementation of a POS tagger using a HMM model. We use a real dataset that has been used in papers as well. Topics that will be demonstrated in this exercise include:

- How to open and preprocess a dataset.
- How do we construct the matrices using numpy.
- Using the matrices for matrix multiplications.
- Follow the pseudo-code of an algorithm to implement it ourselves.

Refer to the .ipynb file for the exercise. This can be opened locally using jupyter notebook, or in a google colab. The latter is easier, since it requires no extra installations on your end.



	SOS	NOUN	DET	VB	ADP	PRON	Total
SOS			II			IIII	6
NOUN		I		III	I		5
DET		IIII					5
VB		I	I		IIII		6
ADP		I	II			II	5
PRON		II		III			5

	SOS	NOUN	DET	VB	ADP	PRON
SOS	IIII		IIII			IIII
The						
I				III		
fly		II		II		
sits					I	
at						
morning		III				
time		II				
my						II
work		I		I		
makes				I		
in					III	
like				I		
breakfast		II				
before					I	
Total	6	10	6	8	5	6

	SOS	I	fly	before	I	work
S	1	0	0	0	0	0
N	0	0	$\frac{4}{9}\times\frac{2}{25}$	0	0	$\frac{1}{25}\times\frac{1}{15}\times\frac{1}{75}$
V	0	0	$\frac{1}{10}$	0	0	$\frac{3}{40}\times\frac{1}{15}\times\frac{1}{75}$
D	0	0	0	0	0	0
A	0	0	0	$\frac{1}{75}$	0	0
P	0	$\frac{4}{9}$	0	0	$\frac{4}{15\times75}$	0

$\frac{2}{5}\times\frac{1}{10}$

$\frac{4}{9}\times\frac{2}{5}\times\frac{2}{10}$

$= \frac{4}{9}\times\frac{2}{5}\times\frac{1}{5}$

$\frac{1}{5}\times\frac{1}{10}\times\frac{4}{6} = \frac{2}{50} = \frac{1}{25}$

$= \frac{1}{50}\times\frac{2}{3} = \frac{1}{75}$

$\frac{4}{9}\times\frac{3}{5}\times\frac{3}{8}$

$= \frac{4}{9}\times\frac{9}{40} = \frac{1}{10}$

$\frac{1}{75}\times\frac{2}{5}\times\frac{4}{6_3}$

$= \frac{4}{15\times75}$

$\frac{3}{40}\times\frac{1}{25}$

$\frac{15}{200}\quad\frac{8}{200}$

$\frac{1}{8}\times\frac{3}{5}$

$= \frac{3}{40}\times\frac{1}{15\times\cancel{75}_{25}}$

$= \frac{1}{25\times15\times40}$

$\frac{1}{25\times}$

Sentence	Model1	Model2	GT
Barack	O	Person	P
Obama	O	Person	P
was	O	O	O
a	O	O	O
great	O	O	O
President	O	Role	R
with	O	O	O
Michelle	O	O	P
Obama	O	O	P
as	O	Person	O
his	O	O	O
First	O	O	R
Lady	O	Role	R
who	O	Role	O
did	O	O	O
important	O	O	O
work	O	Role	O

① Accuracy

Model 1 $\frac{10}{17} = 0.588$

Model 2 $\frac{11}{17} = 0.647$

② F1 for P

Precision = 0

Recall = 0

F1 for R

$P = 0$

$R = 0$

F1 for O

$\text{Precision} = \frac{10}{17}$

$\text{Recall} = \frac{10}{10}$

$\therefore F1_O = \frac{2 \times 0.588}{1.588}$

$= 0.741$

$\text{Macro-F1} = \frac{1}{3} \times 0.741 = 0.247$

F1 for R

$P_R = \frac{2}{4} = \frac{1}{2}$

$R_R = \frac{2}{3} = 0.67$

$F1_R = 0.57$

$\text{Macro}_{F1} = \frac{0.571 + 0.571 + 0.7}{3}$

$= 0.614$

∴ Model 2 is better.

∴ F1-metric should be used.

M1 is just using constant predictions & since this is a class imbalance problem accuracy is misleading.

③ $y = \begin{pmatrix} \frac{1}{0} \end{pmatrix}$

③ F1 for O

$P_O = \frac{7}{10} = 0.7$

$R_O = \frac{7}{10} = 0.7$

$F1_O = \frac{2 \times 0.7 \times 0.7}{2 \times 0.7}$

$= 0.7$

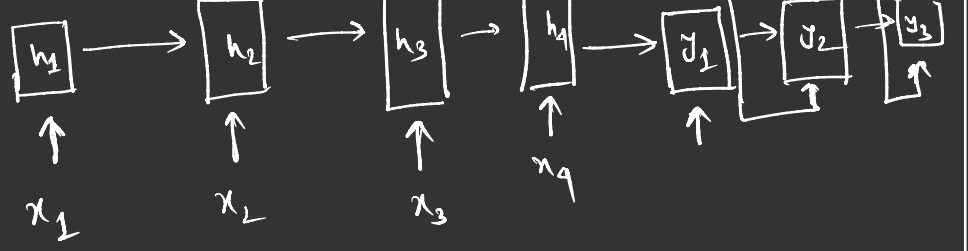
F1 for P

$P_P = \frac{2}{3} = 0.67$

$R_P = \frac{2}{4} = \frac{1}{2} = 0.5$

$F1_R = \frac{2 \times 0.67 \times 0.5}{0.67 + 0.5}$

$= 0.57$



x_1 = vector = $\begin{pmatrix}0\\0\\0\\1\\0\\0\\0\end{pmatrix}$ x_3 = $\begin{pmatrix}0\\0\\1\\0\\0\\0\\0\end{pmatrix}$

x_2 = is = $\begin{pmatrix}1\\0\\0\\0\\0\\0\\0\end{pmatrix}$ x_4 = $\begin{pmatrix}0\\1\\0\\0\\0\\0\\0\end{pmatrix}$

$$\therefore h_0 = W^{(in)T} x_1 + W^{(enc,h)} h_0 + b^{(enc,h)}$$

$$= \begin{pmatrix}0.4\\-0.5\end{pmatrix} + \begin{pmatrix}1\\2\end{pmatrix} = \begin{pmatrix}1.4\\1.5\end{pmatrix}$$

$$\therefore h_2 = \begin{pmatrix}0.3\\-0.2\end{pmatrix} + \begin{pmatrix}1.3 & -0.5\\-0.5 & 0.5\end{pmatrix}\begin{pmatrix}1.4\\1.5\end{pmatrix} + \begin{pmatrix}1\\2\end{pmatrix}$$

$$= \begin{pmatrix}0.3\\-0.2\end{pmatrix} + \begin{pmatrix}1.07\\0.05\end{pmatrix} + \begin{pmatrix}1\\2\end{pmatrix} = \begin{pmatrix}2.37\\1.85\end{pmatrix}$$

$$\therefore h_3 = \begin{pmatrix}0.5\\1\end{pmatrix} + \begin{pmatrix}2.156\\-0.26\end{pmatrix} + \begin{pmatrix}1\\2\end{pmatrix}$$

$$= \begin{pmatrix}3.66\\2.74\end{pmatrix}$$

$$h_4 = \begin{pmatrix}-1.1\\0.8\end{pmatrix} + \begin{pmatrix}3.388\\-0.46\end{pmatrix} + \begin{pmatrix}1\\2\end{pmatrix}$$

$$= \begin{pmatrix}3.28\\2.34\end{pmatrix}$$

Decoder

$$h_0^{dec} = \begin{pmatrix}3.28\\2.34\end{pmatrix}$$

$$\therefore h_1^{dec} = W^{(in)T} \cdot SOS + W^{(dec,h)} h_0^{dec} + b^{(dec,h)}$$

$$= \begin{pmatrix}-1 & 0.1\\0.4 & 0.7\end{pmatrix}\begin{pmatrix}3.28\\2.34\end{pmatrix} + \begin{pmatrix}0.5\\0.\end{pmatrix}$$

$$= \begin{pmatrix}-1.85\\1.97\end{pmatrix}$$

$$\therefore y_1 = W^{(out,pred)T} \cdot h_1^D + b^{(out,pred)}$$

$$= \begin{pmatrix}0.6 & 0.5\\-0.2 & 0.8\end{pmatrix}\begin{pmatrix}-1.85\\1.97\end{pmatrix} + \begin{pmatrix}0.4\\0.8\end{pmatrix}$$

$$= \begin{pmatrix}3.638\\1.216\end{pmatrix}$$

$y_1^{(label)} = \begin{pmatrix}0.08\\0.92\end{pmatrix}$ w_1 = President of

$$\therefore h_2^{dec} = W^{(in)T} \cdot w_1 + W^{(dec,h)T} \cdot h_1^{dec} + b^{(dec,h)}$$

$$= \begin{pmatrix}0.9\\0.5\end{pmatrix} + \begin{pmatrix}-1 & 0.4\\0.1 & 0.7\end{pmatrix}\begin{pmatrix}-1.85\\1.97\end{pmatrix} + \begin{pmatrix}0.5\\0\end{pmatrix}$$

$$= \begin{pmatrix}4.038\\1.694\end{pmatrix}$$

$$\therefore y_2 = softmax\left(W^{(out,emb)T} \cdot h_2^{dec} + b^{(out,emb)}\right)$$

$$= (0,0,0,1,0,0,0) = \text{vector}$$

$$h_3^{dec} = \begin{pmatrix}0.4\\-0.5\end{pmatrix} + \begin{pmatrix}-1 & 0.4\\0.1 & 0.7\end{pmatrix}\begin{pmatrix}4.038\\1.694\end{pmatrix} + \begin{pmatrix}0.5\\0\end{pmatrix}$$

$$= \begin{pmatrix}-2.46\\1.09\end{pmatrix}$$

$$y_3 = softmax\left(W^{(out,emb)T} \cdot h_3^{dec} + b^{(out,emb)}\right)$$

$$= (0,0,0,0,0,1,0)$$