University of Pittsburgh

School of Computing and Information

CS2710

Natural Language Processing

Sequence Labeling

Date: November 03, 2023

Jacob Hoffman

Advisors:

Michael Yoder, PhD

1. POS tagging with an HMM

Consider a Hidden Markov Model with the following parameters: postags = {NOUN, AUX, VERB}, words = {'Patrick', 'Cherry', 'can', 'will', 'see', 'spot'}

Initial probabilities:

	π
NOUN	0.7
AUX	0.1
VERB	0.2

Transition probabilities: The format is P(column_tag | row_tag), e.g. P(AUX | NOUN) = 0.3.

	NOUN	**AUX**	**VERB**
NOUN	0.2	0.3	0.5
AUX	0.4	0.1	0.5
VERB	0.8	0.1	0.1

Emission probabilities:

	Patrick	**Cherry**	**can**	**will**	**see**	**spot**
NOUN	0.3	0.2	0.1	0.1	0.1	0.2
AUX	0	0	0.4	0.6	0	0
VERB	0	0	0.1	0.2	0.5	0.2

Using the Viterbi algorithm and the given HMM, find the most likely tag sequence for the following 2 sentences.

1. "Patrick can see Cherry"
2. "will Cherry spot Patrick"

To get you started on the Viterbi tables, here are the first 2 columns for the first sentence. You'll also want to include the backtraces.

Sentence 1:

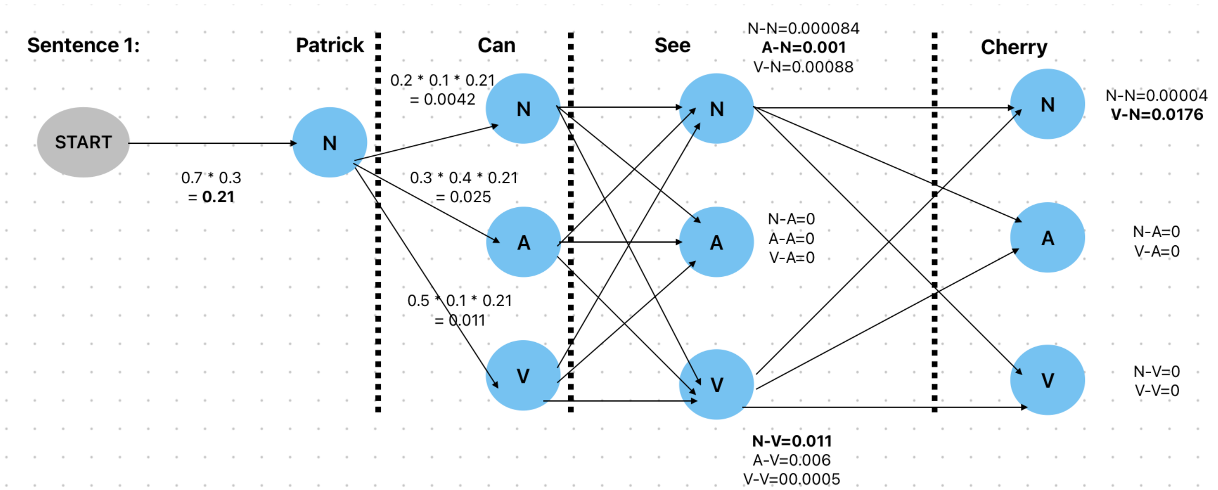
Most likely tag sequence: NOUN, AUX, VERB, NOUN

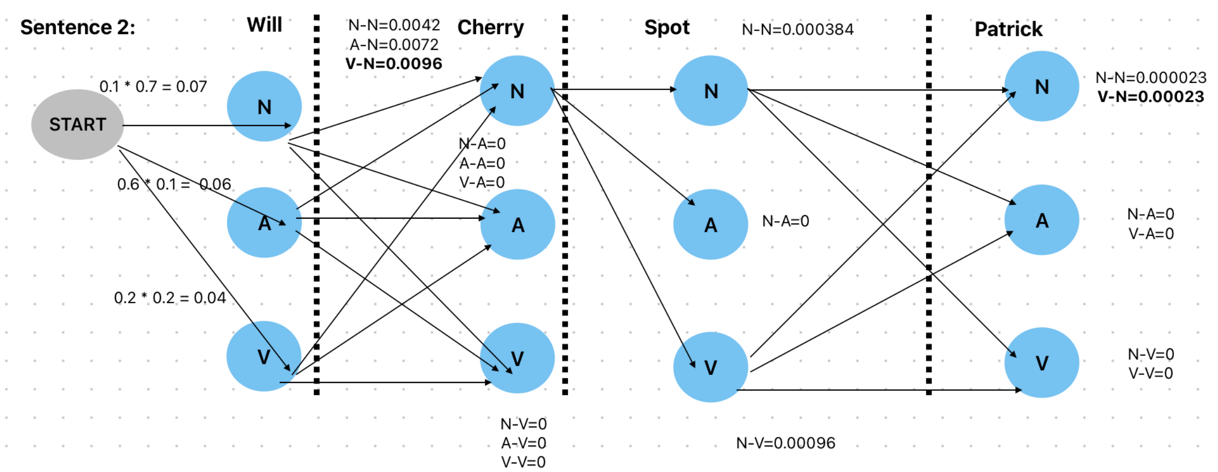
POS state	Patrick	can	see	Cherry
NOUN	0.21	0.0042	0.001	0.0176
AUX	0	0.0252	0	0
VERB	0	0.0105	0.011	0

Sentence 2:

Most likely tag sequence: NOUN, NOUN, VERB, NOUN

POS state	will	Cherry	spot	Patrick
NOUN	0.07	0.0096	0.00038	0.00023
AUX	0.06	0	0	0
VERB	0.04	0	0.00096	0


Sentence 1:
START
Patrick
N
0.7 * 0.3
= 0.21
Can
0.2 * 0.1 * 0.21
= 0.0042
N
0.3 * 0.4 * 0.21
= 0.025
A
0.5 * 0.1 * 0.21
= 0.011
V
See
N-N=0.000084
A-N=0.001
V-N=0.00088
N
N-A=0
A-A=0
V-A=0
A
V
N-V=0.011
A-V=0.006
V-V=00.0005
Cherry
N
N-N=0.00004
V-N=0.0176
A
N-A=0
V-A=0
V
N-V=0
V-V=0


Sentence 2:
START
Will
0.1 * 0.7 = 0.07
N
0.6 * 0.1 = 0.06
A
0.2 * 0.2 = 0.04
V
N-N=0.0042
A-N=0.0072
V-N=0.0096
Cherry
N
N-A=0
A-A=0
V-A=0
A
V
N-V=0
A-V=0
V-V=0
Spot
N-N=0.000384
N
A
N-A=0
V
N-V=0.00096
Patrick
N
N-N=0.000023
V-N=0.00023
A
N-A=0
V-A=0
V
N-V=0
V-V=0

2. Fine-tune BERT-based NER models

In this section, you will fine-tune multiple pretrained BERT-based models on Spanish NER data. Specifically, you will fine-tune at least one model pretrained on masked language modeling (MLM) on Spanish data, and at least one model pretrained on NER in a language other than Spanish.

Copy this [skeleton Colab notebook]() and fill in the places that are specified.

Deliverables for part 2

In your report, include:

1. The F1 score on the CoNLL-2003 Spanish test set for
    1. the model pretrained on MLM in Spanish

chriskhanhtran/spanberta results:

Epoch	Training Loss	Validation Loss	Precision	Recall	F1	Accuracy
1	0.090800	0.136013	0.700184	0.785616	0.740444	0.963927
2	0.053200	0.129263	0.709882	0.798943	0.751784	0.967032
3	0.033100	0.134147	0.742424	0.821921	0.780153	0.969625

TrainOutput(global_step=3123, training_loss=0.06916602880115735, metrics={'train_runtime': 516.687, 'train_samples_per_second': 48.331, 'train_steps_per_second': 6.044, 'total_flos': 901816673340960.0, 'train_loss': 0.06916602880115735, 'epoch': 3.0})

Test example results:

[
{'entity_group': 'PER',
'score': 0.9986547,
'word': ' Miguel Salgado',
'start': 13,
'end': 27},

{'entity_group': 'ORG',
'score': 0.77422214,
'word': ' Universidad de Pit',
'start': 43,
'end': 61},

{'entity_group': 'LOC',
'score': 0.53306746,
'word': 'tsburgh',
'start': 61,
'end': 68},

{'entity_group': 'LOC',
'score': 0.9938329,
'word': ' Pittsburgh.',
'start': 79,
'end': 90}
]

2. the model pretrained on NER in another language

dbmdz/bert-bert-cased-finetuned-conll03-english results:

Epoch	Training Loss	Validation Loss	Precision	Recall	F1	Accuracy
1	0.103100	0.137604	0.690581	0.756434	0.722009	0.960615
2	0.060700	0.132891	0.713867	0.783088	0.746877	0.964555
3	0.030100	0.143020	0.743174	0.800551	0.770796	0.968030

TrainOutput(global_step=3123, training_loss=0.07492402048016297, metrics={'train_runtime': 373.5381, 'train_samples_per_second': 66.853, 'train_steps_per_second': 8.361, 'total_flos': 771434741985264.0, 'train_loss': 0.07492402048016297, 'epoch': 3.0})

Test example results:

[
{'entity_group': 'PER',
'score': 0.99591243,
'word': 'Miguel Salgado. Trabajo',
'start': 13,
'end': 36},

{'entity_group': 'ORG',
'score': 0.8968439,
'word': 'Universidad de Pittsburgh',
'start': 43,
'end': 68},

{'entity_group': 'LOC',
'score': 0.9904759,
'word': 'Pittsburgh',
'start': 79,
'end': 89}
]

2. A brief discussion of which model performs better and any choices you made about hyperparameters in training

They both performed well, but the Spanish model seems to be *slightly* better (based on metrics). However, the test results seem to be better with the English model. I'm not entirely sure why, but I kind of expected the Spanish model to do much better than the English model.

3. A link to your copied and filled out Colab notebook

https://drive.google.com/file/d/1Bou9p7U3zV3RLaCH4-lMkPtoPkWhoHAL/view?usp=sharing