Recitation 1

Viterbi Decoding & HW1

Chaitanya Ahuja

Carnegie Mellon University

Table of contents

Introduction

Task

Named Entity Recognition

I-ORG	O	I-MISC	O	O	O	I-MISC	O
EU	rejects	German	call	to	boycott	British	lamb

HW Goals

HW Goals

- How to extract features, and estimate scores using a linear transform

HW Goals

- How to extract features, and estimate scores using a linear transform
- Use the scores as a metric for Viterbi Decoding

Decoding

Mathematical Formulation

Let $s_1, \ldots, s_n \in \Omega$ and $w_1, \ldots, w_n \in \Sigma$.

$$
\begin{aligned}
score_0 &= \gamma \left(s_1 \middle| < \text{START} > \right) \eta \left(w_1 \middle| s_1 \right) && (1) \\
score_n &= \max_{s_n} \gamma \left(s_n \middle| s_{n-1} \right) \eta \left(w_n \middle| s_n \right) score_{n-1} && (2) \\
&= \max_{s_1, \ldots s_n} \prod_{i=1}^{n} \gamma \left(s_i \middle| s_{i-1} \right) \eta \left(w_i \middle| s_i \right) && (3)
\end{aligned}
$$

Mathematical Formulation

Let $s_1, \ldots, s_n \in \Omega$ and $w_1, \ldots, w_n \in \Sigma$.

$$score_0 = \gamma\left(s_1 \middle| < \text{START} >\right) \eta\left(w_1 \middle| s_1\right) \quad (1)$$

$$score_n = \max_{s_n} \gamma\left(s_n \middle| s_{n-1}\right) \eta\left(w_n \middle| s_n\right) score_{n-1} \quad (2)$$

$$= \max_{s_1,\ldots s_n} \prod_{i=1}^{n} \gamma\left(s_i \middle| s_{i-1}\right) \eta\left(w_i \middle| s_i\right) \quad (3)$$

$$\gamma\left(s_i \middle| s_{i-1}\right) \eta\left(w_i \middle| s_i\right) \approx \exp\left(\mathbf{W}^T g\left(w_i, w_{i-1}, w_{i+1}, s_i, s_{i-1}\right)\right)$$

Mathematical Formulation

Let $s_1, \ldots, s_n \in \Omega$ and $w_1, \ldots, w_n \in \Sigma$.

$$score_0 = \gamma\left(s_1 | < \text{START} >\right) \eta\left(w_1 | s_1\right) \tag{1}$$

$$score_n = \max_{s_n} \gamma\left(s_n | s_{n-1}\right) \eta\left(w_n | s_n\right) score_{n-1} \tag{2}$$

$$= \max_{s_1, \ldots s_n} \prod_{i=1}^{n} \gamma\left(s_i | s_{i-1}\right) \eta\left(w_i | s_i\right) \tag{3}$$

$$\boxed{\gamma\left(s_i | s_{i-1}\right) \eta\left(w_i | s_i\right) \approx \exp\left(\mathbf{W}^T g\left(w_i, w_{i-1}, w_{i+1}, s_i, s_{i-1}\right)\right)}$$

$$score_n = \max_{s_1, \ldots s_n} \prod_{i=1}^{n} \exp\left(\mathbf{W}^T g\left(w_i, w_{i-1}, w_{i+1}, s_i, s_{i-1}\right)\right) \tag{4}$$

$$\log\left(score_n\right) = \max_{s_1, \ldots s_n} \sum_{i=1}^{n} \mathbf{W}^T g\left(w_i, w_{i-1}, w_{i+1}, s_i, s_{i-1}\right) \tag{5}$$

$$\log\left(score_n\right) = \max_{s_n} \mathbf{W}^T g\left(w_n, w_{n-1}, w_{n+1}, s_n, s_{n-1}\right) + \log\left(score_{n-1}\right) \tag{6}$$

Feature Extraction

What are features and weights?

- **Features:** $g\left(w_i, w_{i-1}, w_{i+1}, s_i, s_{i-1}\right) = \sum_{\forall j} g_j$
  The represent the occurrence of a certain set of patterns. For example:
  $g_1(w_i, s_i)$: `Pi=NNP:Ti=I-LOC 1.0`
  $g_2(w_i, w_{i+1}, s_i)$: `Wi=France:Wi+1=and:Ti=I-LOC 1.0`
  $g_3(s_i, s_{i-1})$: `Ti-1=<START>:Ti=I-LOC`
  $\ldots$

What are features and weights?

- **Features:** $g\left(w_i, w_{i-1}, w_{i+1}, s_i, s_{i-1}\right) = \sum_{\forall j} g_j$

  The represent the occurrence of a certain set of patterns. For example:

  $g_1(w_i, s_i)$: `Pi=NNP:Ti=I-LOC 1.0`

  $g_2(w_i, w_{i+1}, s_i)$: `Wi=France:Wi+1=and:Ti=I-LOC 1.0`

  $g_3(s_i, s_{i-1})$: `Ti-1=<START>:Ti=I-LOC`

  ...

- Weights **W** are proportional to the probability of the occurrence of the corresponding features. For example:

```
Oi=rating:Ti=I-LOC -3.0
Oi=october:Ti-1=O:Ti=O 10.0

CAPi=False:Ti=O 50.0
CAPi=False:Ti=O -31.0
```

How do we extract features?

$s_1, \ldots, s_n \in \Omega$ and $w_1, \ldots, w_n \in \Sigma$.

- For each i, features that depend on w_i can be deterministically estimated.

How do we extract features?

$s_1, \ldots, s_n \in \Omega$ and $w_1, \ldots, w_n \in \Sigma$.

- For each i, features that depend on w_i can be deterministically estimated.
- But, features involving s_i are a little more tricky. They must be estimated assuming that all possible states $s_i \in \Omega$ could have occurred and the best state is chosen based on scores for step $i - 1$

Example: How do we extract features?[1]

```
EU NNP I-NP I-ORG
rejects VBZ I-VP O
German JJ I-NP I-MISC
call NN I-NP O
to TO I-VP O
boycott VB I-VP O
British JJ I-NP I-MISC
lamb NN I-NP O
```

[1] The chosen example does not reflect the views of the author but my laziness to go beyond the first sample in the data:P

Example: How do we extract features?[1]

```
EU NNP I-NP I-ORG
rejects VBZ I-VP O
German JJ I-NP I-MISC
call NN I-NP O
to TO I-VP O
boycott VB I-VP O
British JJ I-NP I-MISC
lamb NN I-NP O
```

Consider the word `British`. The three of the many features would be:

[1] The chosen example does not reflect the views of the author but my laziness to go beyond the first sample in the data:P

Example: How do we extract features?[1]

```
EU NNP I-NP I-ORG
rejects VBZ I-VP O
German JJ I-NP I-MISC
call NN I-NP O
to TO I-VP O
boycott VB I-VP O
British JJ I-NP I-MISC
lamb NN I-NP O
```

Consider the word `British`. The three of the many features would be:

- `Pi=JJ:Ti=`$< \forall s_i \in \Omega >$ `1.0`

[1] The chosen example does not reflect the views of the author but my laziness to go beyond the first sample in the data:P

Example: How do we extract features?[1]

```
EU NNP I-NP I-ORG
rejects VBZ I-VP O
German JJ I-NP I-MISC
call NN I-NP O
to TO I-VP O
boycott VB I-VP O
British JJ I-NP I-MISC
lamb NN I-NP O
```

Consider the word `British`. The three of the many features would be:

- `Pi=JJ:Ti=`$< \forall s_i \in \Omega >$ `1.0`
- `Wi=British:Wi+1=lamb:Ti=`$< \forall s_i \in \Omega >$ `1.0`

[1] The chosen example does not reflect the views of the author but my laziness to go beyond the first sample in the data:P

Example: How do we extract features?[1]

```
EU NNP I-NP I-ORG
rejects VBZ I-VP O
German JJ I-NP I-MISC
call NN I-NP O
to TO I-VP O
boycott VB I-VP O
British JJ I-NP I-MISC
lamb NN I-NP O
```

Consider the word `British`. The three of the many features would be:

- `Pi=JJ:Ti=`$< \forall s_i \in \Omega >$ `1.0`
- `Wi=British:Wi+1=lamb:Ti=`$< \forall s_i \in \Omega >$ `1.0`
- `Ti-1=`$< \forall s_i \in \Omega >$`:Ti=`$< \forall s_i \in \Omega >$ `1.0`

[1] The chosen example does not reflect the views of the author but my laziness to go beyond the first sample in the data:P

How do we estimate weights?

Estimation: Wait for a couple of weeks.

For this Assignment, we just need to use the given weights.

How do we estimate weights?

We use the extracted features and find the corresponding weights $\forall s_i, s_{i-1} \in \Omega$ and call this matrix $\mathbf{PScore}_i$.

s_i/s_{i-1}	O	I-PER	I-ORG	I-MISC	I-LOC	B-ORG	B-MISC	B-LOC
O	25.0	5.0	1.0	-3.0	-2.0	-4.0	-7.0	-2.0
I-PER	6.0	23.0	-9.0	-7.0	-15.0	0.0	-1.0	-1.0
I-ORG	2.0	-16.0	32.0	-9.0	-13.0	-1.0	-1.0	0.0
I-MISC	-4.0	-9.0	-11.0	23.0	-2.0	0.0	6.0	-1.0
I-LOC	2.0	-4.0	-17.0	-8.0	25.0	0.0	-1.0	2.0
B-ORG	-2.0	-2.0	2.0	-1.0	0.0	3.0	0.0	0.0
B-MISC	-6.0	-1.0	-2.0	8.0	-2.0	0.0	0.0	0.0
B-LOC	-5.0	-1.0	-2.0	-1.0	8.0	0.0	0.0	0.0

How do we estimate scores?

$s_1, \ldots, s_n \in \Omega$ and $w_1, \ldots, w_n \in \Sigma$.

	s_{i-1}
O	$\text{Score}_{i-1}(1)$
I-PER	$\text{Score}_{i-1}(2)$
I-ORG	$\text{Score}_{i-1}(3)$
I-MISC	$\text{Score}_{i-1}(4)$
I-LOC	$\text{Score}_{i-1}(5)$
B-ORG	$\text{Score}_{i-1}(6)$
B-MISC	$\text{Score}_{i-1}(7)$
B-LOC	$\text{Score}_{i-1}(8)$

s_i / s_{i-1}	O	I-PER	I-ORG	I-MISC	I-LOC	B-ORG	B-MISC	B-LOC
O	25.0	5.0	1.0	-3.0	-2.0	-4.0	-7.0	-2.0
I-PER	6.0	23.0	-9.0	-7.0	-15.0	0.0	-1.0	-1.0
I-ORG	2.0	-16.0	32.0	-9.0	-13.0	-1.0	-1.0	0.0
I-MISC	-4.0	-9.0	-11.0	23.0	-2.0	0.0	6.0	-1.0
I-LOC	2.0	-4.0	-17.0	-8.0	25.0	0.0	-1.0	2.0
B-ORG	-2.0	-2.0	2.0	-1.0	0.0	3.0	0.0	0.0
B-MISC	-6.0	-1.0	-2.0	8.0	-2.0	0.0	0.0	0.0
B-LOC	-5.0	-1.0	-2.0	-1.0	8.0	0.0	0.0	0.0

Figure 1: Score_{i-1} (Left) and PScore_i (Right)

- For each i, estimate the best local score by considering all possible states for s_i and s_{i-1}. This is an $O(|\Omega|^2)$ operation.

How do we estimate scores?

$s_1, \ldots, s_n \in \Omega$ and $w_1, \ldots, w_n \in \Sigma$.

	s_{i-1}
O	$\text{Score}_{i-1}(1)$
I-PER	$\text{Score}_{i-1}(2)$
I-ORG	$\text{Score}_{i-1}(3)$
I-MISC	$\text{Score}_{i-1}(4)$
I-LOC	$\text{Score}_{i-1}(5)$
B-ORG	$\text{Score}_{i-1}(6)$
B-MISC	$\text{Score}_{i-1}(7)$
B-LOC	$\text{Score}_{i-1}(8)$

s_i / s_{i-1}	O	I-PER	I-ORG	I-MISC	I-LOC	B-ORG	B-MISC	B-LOC
O	25.0	5.0	1.0	-3.0	-2.0	-4.0	-7.0	-2.0
I-PER	6.0	23.0	-9.0	-7.0	-15.0	0.0	-1.0	-1.0
I-ORG	2.0	-16.0	32.0	-9.0	-13.0	-1.0	-1.0	0.0
I-MISC	-4.0	-9.0	-11.0	23.0	-2.0	0.0	6.0	-1.0
I-LOC	2.0	-4.0	-17.0	-8.0	25.0	0.0	-1.0	2.0
B-ORG	-2.0	-2.0	2.0	-1.0	0.0	3.0	0.0	0.0
B-MISC	-6.0	-1.0	-2.0	8.0	-2.0	0.0	0.0	0.0
B-LOC	-5.0	-1.0	-2.0	-1.0	8.0	0.0	0.0	0.0

Figure 1: Score_{i-1} (Left) and PScore_i (Right)

- $\text{Score}_i(j) = \max_k \text{Score}_{i-1}(k) + PScore_i(j,k) \qquad \forall j \in \{1, \ldots |\Omega|\}$

Caveats

Caveat 1: Features do not depend on states

	s_{i-1}
O	c_{i-1}
I-PER	c_{i-1}
I-ORG	c_{i-1}
I-MISC	c_{i-1}
I-LOC	c_{i-1}
B-ORG	c_{i-1}
B-MISC	c_{i-1}
B-LOC	c_{i-1}

	s_i
O	c_i
I-PER	c_i
I-ORG	c_i
I-MISC	c_i
I-LOC	c_i
B-ORG	c_i
B-MISC	c_i
B-LOC	c_i

- Scores are constant across States

Caveat 1: Features do not depend on states

	s_{i-1}
O	c_{i-1}
I-PER	c_{i-1}
I-ORG	c_{i-1}
I-MISC	c_{i-1}
I-LOC	c_{i-1}
B-ORG	c_{i-1}
B-MISC	c_{i-1}
B-LOC	c_{i-1}

	s_i
O	c_i
I-PER	c_i
I-ORG	c_i
I-MISC	c_i
I-LOC	c_i
B-ORG	c_i
B-MISC	c_i
B-LOC	c_i

- Scores are constant across States
- Which makes the prediction of states Random!

Caveat 2: Features only depend on the current state (s_i)

	s_{i-1}	
O	$\text{Score}_{i-1}(1)$	
I-PER	$\text{Score}_{i-1}(2)$	
I-ORG	$\text{Score}_{i-1}(3)$	
I-MISC	$\text{Score}_{i-1}(4)$	
I-LOC	$\text{Score}_{i-1}(5)$	
B-ORG	$\text{Score}_{i-1}(6)$	
B-MISC	$\text{Score}_{i-1}(7)$	
B-LOC	$\text{Score}_{i-1}(8)$	

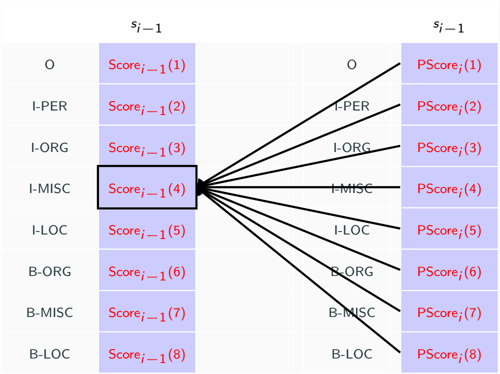
s_i / s_{i-1}	O	I-PER	...
O	$\text{PScore}_i(1, 1)$	$\text{PScore}_i(1, 2)$	...
I-PER	$\text{PScore}_i(2, 1)$	$\text{PScore}_i(2, 2)$	...
I-ORG	$\text{PScore}_i(3, 1)$	$\text{PScore}_i(3, 2)$	...
I-MISC	$\text{PScore}_i(4, 1)$	$\text{PScore}_i(4, 2)$	...
I-LOC	$\text{PScore}_i(5, 1)$	$\text{PScore}_i(5, 2)$	...
B-ORG	$\text{PScore}_i(6, 1)$	$\text{PScore}_i(6, 2)$	...
B-MISC	$\text{PScore}_i(7, 1)$	$\text{PScore}_i(7, 2)$	...
B-LOC	$\text{PScore}_i(8, 1)$	$\text{PScore}_i(8, 2)$	...

Caveat 2: Features only depend on the current state (s_i)

	s_{i-1}				s_{i-1}
O	$\text{Score}_{i-1}(1)$			O	$\text{PScore}_i(1)$
I-PER	$\text{Score}_{i-1}(2)$			I-PER	$\text{PScore}_i(2)$
I-ORG	$\text{Score}_{i-1}(3)$			I-ORG	$\text{PScore}_i(3)$
I-MISC	$\text{Score}_{i-1}(4)$			I-MISC	$\text{PScore}_i(4)$
I-LOC	$\text{Score}_{i-1}(5)$			I-LOC	$\text{PScore}_i(5)$
B-ORG	$\text{Score}_{i-1}(6)$			B-ORG	$\text{PScore}_i(6)$
B-MISC	$\text{Score}_{i-1}(7)$			B-MISC	$\text{PScore}_i(7)$
B-LOC	$\text{Score}_{i-1}(8)$			B-LOC	$\text{PScore}_i(8)$

- The PScore_i reduces to a 1-D matrix.

Caveat 2: Features only depend on the current state (s_i)



- The PScore_i reduces to a 1-D matrix.
- It is equivalent to a greedy approach.

Caveat 3: Features depend on more than 2 states $(s_i, s_{i-1} \ldots, s_{i-k})$

- Adding dependency on far-away states (s_{i-k} for $k \geq 2$) changes

Caveat 3: Features depend on more than 2 states $(s_i, s_{i-1} \ldots, s_{i-k})$

- Adding dependency on far-away states (s_{i-k} for $k \geq 2$) changes
    - Computational complexity to $\mathrm{O}\left(n|\Omega|^{k+1}\right)$

Caveat 3: Features depend on more than 2 states $(s_i, s_{i-1} \ldots, s_{i-k})$

- Adding dependency on far-away states (s_{i-k} for $k \geq 2$) changes
    - Computational complexity to $O\left(n|\Omega|^{k+1}\right)$
    - Space complexity to $O\left(n|\Omega|^{k}\right)$

Caveat 3: Features depend on more than 2 states $(s_i, s_{i-1} \ldots, s_{i-k})$

- Adding dependency on far-away states (s_{i-k} for $k \geq 2$) changes
  - Computational complexity to $\mathrm{O}\left(n|\Omega|^{k+1}\right)$
  - Space complexity to $\mathrm{O}\left(n|\Omega|^{k}\right)$
- Hence, if $|\Omega| = m$, the matrix size to store the relevant scores for features dependent on (s_i, s_{i-1}, s_{i-2}) is $n \times m \times m$

Questions ?