Advanced Human Language Technologies
Exercises on features for log-linear models

Features for classification

Exercise 1.

Given a vocabulary V with $|V| = 1000$, and a word history $x = \langle x_1, \ldots, x_{i-1} \rangle$, and assuming the size of our alphabet is 26 letters, we define the following feature template for any word u and 4-letter suffix s:

$$\mathbf{f}_{u,s}(x, y) = \begin{cases} 1 & \text{if } y = u \text{ and } x_{i-1} \text{ ends with } s \\ 0 & \text{otherwise} \end{cases}$$

1. How many possible features are there in this model?
   (a) 1000×26^4
   (b) 26^4
   (c) 1000
   (d) 1000×1000
2. Now consider a training set of size $n = 10,000$. Which is a good upper bound on the number of features introduced by this training set?
   (a) 1000×26^4
   (b) 10000×26^4
   (c) 1000
   (d) 10000

SOLUTION

1. The answer is (d). The potentially existing features are all combinations of a word in V with the 4-letter suffix of the previous word. Since the previous word will be also one of the $1,000$ in $|V|$, there are at most $1,000$ different observable 4-letter suffixes. Thus, the number of potential features is $1,000 \times 1,000$.
2. The answer is (d). Even the training set has size $10,000$, there are only $1,000$ different possible words we can observe. Since the data set has size $10,000$, there will be repetitions of the same word. If all occurrences of the same word have a different previous word, each word in the training set would generate a different feature (since the suffix of the previous word would be different). Thus, we will get at most $10,000$ different features from this training data.

Exercise 2.

Consider the label set $\mathcal{Y}$ consisting of part-of-speech tags with $|\mathcal{Y}| = 50$, and the set $\mathcal{X}$ consisting of histories of the form $\langle t_1, \ldots, t_i, w_1, \ldots, w_n, i \rangle$. Also assume we have a vocabulary V of size 1000. We define the following feature template for any word $u \in V$ and PoS-tag $t \in \mathcal{Y}$:

$$\mathbf{f}_{u,t}(x, y) = \begin{cases} 1 & \text{if } w_i = u \text{ and } y = t \\ 0 & \text{otherwise} \end{cases}$$

1. How many possible features are there in this model?
2. How many different features are introduced by the training set consisting of the following word/tag pairs ?
   `the/DT man/NN fishes/V for/ADP the/DT fishes/NN`

SOLUTION

1. The given pattern can generate a different feature for each combination of a word in V and a tag in $\mathcal{Y}$. Thus, the number of potential features is $1,000 \times 50 = 50,000$. However, the actual number will be much smaller, since each word will not appear with all possible PoS tags, but just with a few of them.
2. The given sentence introduces 5 different features:

   $\mathbf{f}_{\text{the,DT}}$
   $\mathbf{f}_{\text{man,NN}}$
   $\mathbf{f}_{\text{fishes,V}}$
   $\mathbf{f}_{\text{for,ADP}}$
   $\mathbf{f}_{\text{fishes,NN}}$

Exercise 3.

Consider the label set $\mathcal{Y} = \{cat, dog, rat, cow\}$ with three simple features:

$$\mathbf{f}_1(x,y) = \begin{cases} 1 & \text{if } x = \texttt{the} \text{ and } y \text{ ends with } \texttt{at} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{f}_2(x,y) = \begin{cases} 1 & \text{if } x = \texttt{the} \text{ and } y \text{ starts with } \texttt{c} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{f}_3(x,y) = \begin{cases} 1 & \text{if } x = \texttt{the} \text{ and } y \text{ has second letter } \texttt{o} \\ 0 & \text{otherwise} \end{cases}$$

Say we are given the weight vector $\langle w_1, w_2, w_3 \rangle = \langle 1, 2, 3 \rangle$.

1. What is the score of the following pairs?
   - $(x, y) = (\texttt{the}, \texttt{cat})$
   - $(x, y) = (\texttt{the}, \texttt{dog})$
   - $(x, y) = (\texttt{the}, \texttt{rat})$
   - $(x, y) = (\texttt{the}, \texttt{cow})$
2. What is the probability of each label for $x = \texttt{the}$?
   - $P(\texttt{cat}|\texttt{the}; w)$ =??
   - $P(\texttt{dog}|\texttt{the}; w)$ =??
   - $P(\texttt{rat}|\texttt{the}; w)$ =??
   - $P(\texttt{cow}|\texttt{the}; w)$ =??
3. What would be the probabilities if the weight vector was:
   - $\langle w_1, w_2, w_3 \rangle = \langle 0, 0, 0 \rangle$.
   - $\langle w_1, w_2, w_3 \rangle = \langle 3, 1, 1 \rangle$.

SOLUTION

1. The scores for given pairs are:
   - $(x, y) = (\texttt{the}, \texttt{cat})$
     This pair satisfies features $\mathbf{f}_1$ and $\mathbf{f}_2$, thus its score is $\mathbf{w} \cdot \mathbf{f} = (w_1, w_2, w_3) \cdot (1, 1, 0) = 1 * 1 + 2 * 1 + 3 * 0 = 3$.
   - $(x, y) = (\texttt{the}, \texttt{dog})$
     This pair satisfies feature $\mathbf{f}_3$, thus its score is $\mathbf{w} \cdot \mathbf{f} = (w_1, w_2, w_3) \cdot (0, 0, 1) = 1*0+2*0+3*1 = 3$.
   - $(x, y) = (\texttt{the}, \texttt{rat})$
     This pair satisfies feature $\mathbf{f}_1$, thus its score is $\mathbf{w} \cdot \mathbf{f} = (w_1, w_2, w_3) \cdot (1, 0, 0) = 1*1+2*0+3*0 = 1$.
   - $(x, y) = (\texttt{the}, \texttt{cow})$
     This pair satisfies features $\mathbf{f}_2$ and $\mathbf{f}_3$, thus its score is $\mathbf{w} \cdot \mathbf{f} = (w_1, w_2, w_3) \cdot (0, 1, 1) = 1 * 0 + 2 * 1 + 3 * 1 = 5$.
2. The probability of each label y is:

$$P(y|x) = \frac{\exp(\mathbf{w} \cdot \mathbf{f}(x, y))}{Z(x)} = \frac{\exp(\mathbf{w} \cdot \mathbf{f}(x, y))}{\sum_{k \in \mathcal{Y}} \exp(\mathbf{w} \cdot \mathbf{f}(x, k))}$$

   Using the scores obtained in the previous exercise, we can compute $Z(x)$:

$$Z(x) = \sum_{k \in \mathcal{Y}} \exp(\mathbf{w} \cdot \mathbf{f}(x, k)) = \exp(3) + \exp(3) + \exp(1) + \exp(5) = 191.3$$

   Thus, the requested probabilities are:

- $P(\texttt{cat}|\texttt{the}; w) = \exp(3)/Z(x) = 0.105$
- $P(\texttt{dog}|\texttt{the}; w) = \exp(3)/Z(x) = 0.105$
- $P(\texttt{rat}|\texttt{the}; w) = \exp(1)/Z(x) = 0.014$
- $P(\texttt{cow}|\texttt{the}; w) = \exp(5)/Z(x) = 0.776$

3. With the modified vectors, the probabilities would be the following:

   - $\langle w_1, w_2, w_3 \rangle = \langle 0, 0, 0 \rangle$.
     If all weigths are zero, the score for any pair (x, y) will be $\mathbf{w} \cdot \mathbf{f} = (0, 0, 0) \cdot \mathbf{f} = 0$.
     Thus, $Z(x) = \exp(0) + \exp(0) + \exp(0) + \exp(0) = 1 + 1 + 1 + 1 = 4$.
     Then, the probability for any class $y \in \mathcal{Y}$ will be:

     $$P(y|x) = \frac{\exp(0)}{Z(x)} = \frac{1}{4} = 0.25$$

     that is, we get a uniform distribution.
   - $\langle w_1, w_2, w_3 \rangle = \langle 3, 1, 1 \rangle$.
     We need to recompute the score for each class:
     - $(x, y) = (\texttt{the}, \texttt{cat})$
       This pair satisfies features $\mathbf{f}_1$ and $\mathbf{f}_2$, thus its score is $\mathbf{w} \cdot \mathbf{f} = (w_1, w_2, w_3) \cdot (1, 1, 0) = 3 * 1 + 1 * 1 + 1 * 0 = 4$.
     - $(x, y) = (\texttt{the}, \texttt{dog})$
       This pair satisfies feature $\mathbf{f}_3$, thus its score is $\mathbf{w} \cdot \mathbf{f} = (w_1, w_2, w_3) \cdot (0, 0, 1) = 3 * 0 + 1 * 0 + 1 * 1 = 1$.
     - $(x, y) = (\texttt{the}, \texttt{rat})$
       This pair satisfies feature $\mathbf{f}_1$, thus its score is $\mathbf{w} \cdot \mathbf{f} = (w_1, w_2, w_3) \cdot (1, 0, 0) = 3 * 1 + 1 * 0 + 1 * 0 = 3$.
     - $(x, y) = (\texttt{the}, \texttt{cow})$
       This pair satisfies features $\mathbf{f}_2$ and $\mathbf{f}_3$, thus its score is $\mathbf{w} \cdot \mathbf{f} = (w_1, w_2, w_3) \cdot (0, 1, 1) = 3 * 0 + 1 * 1 + 1 * 1 = 2$.

     Using these scores, we can compute $Z(x)$:

     $$Z(x) = \sum_{k \in \mathcal{Y}} \exp(\mathbf{w} \cdot \mathbf{f}(x, k)) = \exp(4) + \exp(1) + \exp(3) + \exp(2) = 84.79$$

     Thus, the requested probabilities are:
     - $P(\texttt{cat}|\texttt{the}; w) = \exp(4)/Z(x) = 0.644$
     - $P(\texttt{dog}|\texttt{the}; w) = \exp(1)/Z(x) = 0.032$
     - $P(\texttt{rat}|\texttt{the}; w) = \exp(3)/Z(x) = 0.237$
     - $P(\texttt{cow}|\texttt{the}; w) = \exp(2)/Z(x) = 0.087$

Exercise 4.

Think of named entities, like *Michael Jackson*, *Barcelona*, or *United Nations*. We can distinguish between different types of entities. For example, *Michael Jackson* is a PERSON, *Barcelona* is a LOCATION and *United Nations* is an ORGANIZATION. Like in all NLP problems, we will frequently encounter difficult cases, such as

```
[ Jack London ] went to Paris.
```

where *Jack London* is a PERSON rather than a LOCATION despite containing a well known location name. There are cases of ambiguity, such as

```
[ Spain ] won the championship.
```

where *Spain* is an ORGANIZATION because in this case it refers to a team or country, not a geographical location. Hence we would like to use linear classification methods that predict the most appropriate class. Let Y be the set of named entity classes (three in our example). In this exercise **we will assume that a previous process has identified the boundaries of named entities, and our goal is just to classify them**. We will make the following linear predictions

$$\text{ne\_type}(x_{1:n}, i, j) = \underset{y \in Y}{\text{argmax}}\, \mathbf{w} \cdot \mathbf{f}(x_{1:n}, i, j, y)$$

where

- $x_{1:n}$ is a sentence with n tokens (x_i is the i-th token)
- i and j are the first and last positions of the entity; thus $x_{i:j}$ is the complete entity
- y is one of the entity types
- $\mathbf{f}(x_{1:n}, i, j, y)$ is a function that returns a feature vector
- $\mathbf{w}$ is a vector of parameters, with one weight per feature

We are interested in creating a set of features for the classifier. For example, feature 1 may indicate if the current entity is the single-word entity *London* and is tagged as LOC (location):

$$\mathbf{f}_1(x_{1:n}, i, j, y) = \begin{cases} 1 & \text{if } i = j \text{ and } x_i = \texttt{London} \text{ and } y = \text{LOC} \\ 0 & \text{otherwise} \end{cases}$$

Rather than specifying each dimension explicitly, we will create *feature templates* that generate a number of features mechanically. Each feature template is identified by a type. For example, the template with type 1 may be

$$\mathbf{f}_{1,l,a}(x_{1:n}, i, j, y) = \begin{cases} 1 & \text{if } i = j \text{ and } x_i = a \text{ and } y = l \\ 0 & \text{otherwise} \end{cases}$$

and will generate an actual feature for every $l \in Y$ and every word $a \in V$ (where V is the set of words in our language). If we set $l = \text{LOC}$ and $a = \text{London}$ we obtain the feature above. In general, this template will create feature dimensions identified by a tuple of the form $\langle 1, l, a \rangle$. We will assume that the parameter vector $\mathbf{w}$ is indexed with the same type of tuple dimensions. For example, $\mathbf{w}_{1,\text{LOC},\text{London}}$ is the parameter associated with the feature above.

4.1. Feature Templates

Write feature templates that capture the following information. For each template, specify the tuple that identifies the feature dimensions. Justify your answers if necessary

- Type 1: The entity is word a
  Answer:

$$\mathbf{f}_{1,l,a}(x_{1:n}, i, j, y) = \begin{cases} 1 & \text{if } i = j \text{ and } x_i = a \text{ and } y = l \\ 0 & \text{otherwise} \end{cases}$$

- Type 2: All entity words are capitalized.
- Type 3: The entity contains word a.
- Type 4: The entity has at least three words, the first is a and the second is b.
- Type 5: Word a appears right before or right after the entity.

4.2. Feature Vectors

Assume we define a feature function with the templates 1 to 5 above. Assume also that our vocabulary V includes all English words. For each case below, write the names of the features that are non-zero:

1. $x_{1:5} =$ `I live in Barcelona .`
   $\mathbf{f}(x_{1:5}, 4, 4, \text{LOC}) = \mathbf{f}_{1,\text{LOC,Barcelona}}, \ldots$
2. $x_{1:12} =$ `Smith wrote in the Journal of the Royal College of Physics .`
   $\mathbf{f}(x_{1:12}, 1, 1, \text{PER}) =$?
   $\mathbf{f}(x_{1:12}, 5, 11, \text{ORG}) =$?
3. $x_{1:6} =$ `Spain won the World Cup .`
   $\mathbf{f}(x_{1:6}, 1, 1, \text{ORG}) =$?
   $\mathbf{f}(x_{1:6}, 1, 1, \text{LOC}) =$?
   $\mathbf{f}(x_{1:6}, 4, 5, \text{ORG}) =$?

4.3. Feature Weights

We will now play the role of a learning algorithm by setting the weights of features. The main point is to show that, while feature templates generate a large number of actual features, a learning algorithm should be able to select a few features which are highly discriminative.

Let's start with a simple example. Assume our training set has two examples:

`[ Barcelona ]`$_{\text{LOC}}$ `is a beautiful city .`
`[ Barcelona ]`$_{\text{ORG}}$ `won the game .`

Assume we use feature templates 1–5 defined above. We need to set some weights such that the training set is perfectly classified, and such that the norm of the weight vector is not too large (in this exercise we will make training error to be zero, and try to set a few non-zero weights). A possible weight vector could be:

$$w_{1,\text{LOC,Barcelona}} = 1$$
$$w_{5,\text{ORG,won}} = 2$$

Thus, with these weights, we have "learned" that `Barcelona` is a location, but that words surrounded by `won` should have a higher positive score for being organizations. Note that this knowledge is acquired via learning: the feature templates do not have prior knowledge about what words are locations or surround organizations.

Assume the following training set:

`[ Maria ]`$_{\text{PER}}$ `is beautiful .`
`[ Barcelona ]`$_{\text{LOC}}$ `is beautiful .`
`[ Jack London ]`$_{\text{PER}}$ `is nice .`
`[ Milan ]`$_{\text{LOC}}$ `is nice .`
`[ Jack Smith ]`$_{\text{PER}}$ `lives in [ Great Yarmouth ]`$_{\text{LOC}}$ `.`
`[ Maria Domenech ]`$_{\text{PER}}$ `works in [ Barcelona ]`$_{\text{LOC}}$ `.`
`[ Maria Muschatta ]`$_{\text{PER}}$ `attends school in [ Santa Maria Della Mole ]`$_{\text{LOC}}$ `.`
`[ Barcelona ]`$_{\text{ORG}}$ `won [ Milan ]`$_{\text{ORG}}$ `.`

Question: Using feature templates 1–5, set a parameter vector with as few features as possible that correctly classifies all examples in the training set.

SOLUTION

Feature Templates

- Type 1: The entity is word a:

$$\mathbf{f}_{1,l,a}(x_{1:n}, i, j, y) = \begin{cases} 1 & \text{if } i = j \text{ and } x_i = a \text{ and } y = l \\ 0 & \text{otherwise} \end{cases}$$

- Type 2: All entity words are capitalized:

$$\mathbf{f}_{2,l}(x_{1:n}, i, j, y) = \begin{cases} 1 & \text{if } \forall k : i \leq k \leq j : capitalized(x_k) \text{ and } y = l \\ 0 & \text{otherwise} \end{cases}$$

- Type 3: The entity contains word a:

$$\mathbf{f}_{3,l,a}(x_{1:n}, i, j, y) = \begin{cases} 1 & \text{if } \exists k : i \leq k \leq j : x_k = a \text{ and } y = l \\ 0 & \text{otherwise} \end{cases}$$

- Type 4: The entity has at least three words, the first is a and the second is b:

$$\mathbf{f}_{4,l,a,b}(x_{1:n}, i, j, y) = \begin{cases} 1 & j \geq i + 2 \text{ and } x_i = a \text{ and } x_{i+1} = b \text{ and } y = l \\ 0 & \text{otherwise} \end{cases}$$

- Type 5: Word a appears right before or right after the entity:

$$\mathbf{f}_{5,l,a}(x_{1:n}, i, j, y) = \begin{cases} 1 & (x_{i-1} = a \text{ or } x_{j+1} = a) \text{ and } y = l \\ 0 & \text{otherwise} \end{cases}$$

Feature Vectors

1. $x_{1:5} =$ `I live in Barcelona .`
   $\mathbf{f}(x_{1:5}, 4, 4, \text{LOC}) = \langle\, \mathbf{f}_{1,\text{LOC,Barcelona}};\ \mathbf{f}_{2,\text{LOC}};\ \mathbf{f}_{3,\text{LOC,Barcelona}};\ \mathbf{f}_{5,\text{LOC,in}};\ \mathbf{f}_{5,\text{LOC,.}} \,\rangle$
2. $x_{1:12} =$ `Smith wrote in the Journal of the Royal College of Physics .`
   $\mathbf{f}(x_{1:12}, 1, 1, \text{PER}) = \langle\, \mathbf{f}_{1,\text{PER,Smith}};\ \mathbf{f}_{2,\text{PER}};\ \mathbf{f}_{3,\text{PER,Smith}};\ \mathbf{f}_{5,\text{PER,wrote}} \,\rangle$
   $\mathbf{f}(x_{1:12}, 5, 11, \text{ORG}) = \langle\, \mathbf{f}_{3,\text{ORG,Journal}};\ \mathbf{f}_{3,\text{ORG,of}};\ \mathbf{f}_{3,\text{ORG,the}};\ \mathbf{f}_{3,\text{ORG,Royal}};\ \mathbf{f}_{3,\text{ORG,College}};$
   $\mathbf{f}_{3,\text{ORG,Physics}};\ \mathbf{f}_{4,\text{ORG,Journal,of}};\ \mathbf{f}_{5,\text{ORG,the}};\ \mathbf{f}_{5,\text{ORG,.}} \,\rangle$
3. $x_{1:6} =$ `Spain won the World Cup .`
   $\mathbf{f}(x_{1:6}, 1, 1, \text{ORG}) = \langle\, \mathbf{f}_{1,\text{ORG,Spain}};\ \mathbf{f}_{2,\text{ORG}};\ \mathbf{f}_{3,\text{ORG,Spain}};\ \mathbf{f}_{5,\text{ORG,won}} \,\rangle$
   $\mathbf{f}(x_{1:6}, 1, 1, \text{LOC}) = \langle\, \mathbf{f}_{1,\text{LOC,Spain}};\ \mathbf{f}_{2,\text{LOC}};\ \mathbf{f}_{3,\text{LOC,Spain}};\ \mathbf{f}_{5,\text{LOC,won}} \,\rangle$
   $\mathbf{f}(x_{1:6}, 4, 5, \text{ORG}) = \langle\, \mathbf{f}_{2,\text{ORG}};\ \mathbf{f}_{3,\text{ORG,World}};\ \mathbf{f}_{3,\text{ORG,Cup}};\ \mathbf{f}_{5,\text{ORG,the}};\ \mathbf{f}_{5,\text{ORG,.}} \,\rangle$

Feature Weights

A parameter vector that correcly classifies all given examples is:

$w_{2,\text{LOC}} = 1$	Default. Any entity (with all words capitalized) has score 1 for being a location.
$w_{3,\text{PER,Maria}} = 2$	If the entity contains *Maria* or *Jack*, the default LOC is overriden with a higher
$w_{3,\text{PER,Jack}} = 2$	score for PER. Note that this fails for *Santa Maria Della Mole*.
$w_{5,\text{LOC,in}} = 3$	Higher score for LOC if word *in* is near. This fixes *Santa Maria Della Mole*.
$w_{5,\text{ORG,won}} = 2$	Higher score for ORG if word *won* is near. This overrides the default LOC for *Barcelona* and *Milan* in the last sentence.

Features for bigram-factored sequence annotation models

Recall the factored linear models for sequence prediction, and think of a named entity task. A bigram-factored sequence model computes:

$$\begin{aligned} tags(x_{1:n}) &= \operatorname*{argmax}_{y_{1:n} \in \mathcal{Y}^n} \mathbf{w} \cdot \mathbf{f}(x, y_{1:n}) \\ &= \operatorname*{argmax}_{y_{1:n} \in \mathcal{Y}^n} \sum_{i=1}^{n} \mathbf{w} \cdot \mathbf{f}(x, i, y_{i-1}, y_i) \end{aligned} \quad (1)$$

where $x_{1:n}$ is an input sentence of n tokens (x_i is the i-th token), $y_{1:n}$ is an output sequence of n tags ($\mathcal{Y}$ is the set of valid tags). $\mathbf{f}(x, i, y_{i-1}, \mathbf{y}_i)$ is a function returning a feature vector of the bigram y_{i-1}, y_i at position i of the sentence (assume that y_0 is a special tag START *that indicates the start of the sequence). $\mathbf{w}$ is a vector of parameters of the same dimensionality of the feature vectors.*

Exercise 5.

We specify features using templates. For example, the following template captures the current word and the current tag:

$$\mathbf{f}_{1,l,a}(x, i, y_{i-1}, y_i) = \begin{cases} 1 & \text{if } x_i = a \text{ and } y_i = l \\ 0 & \text{otherwise} \end{cases}$$

Write feature templates that capture the following patterns. Justify your answers if necessary.

- $\mathbf{f}_{2,a}$: the current word is the first of the sentence, it is capitalized, and its tag is a
- $\mathbf{f}_{3,s,a}$: 3-letter prefix of the current word, together with the current tag
- $\mathbf{f}_{4,w,a,b}$: the current word, the current tag, and the previous tag
- $\mathbf{f}_{5,w,v,a}$: the two previous words and the current tag
- $\mathbf{f}_{6,a,b,c}$: the two previous tags and the current tag

SOLUTION

$$\mathbf{f}_{2,a}(x_{1:n}, i, y_{i-1}, y_i) = \begin{cases} 1 & \text{if } i = 1 \text{ and } capitalized(x_i) \text{ and } y_i = a \\ 0 & otherwise \end{cases}$$

$$\mathbf{f}_{3,s,a}(x_{1:n}, i, y_{i-1}, y_i) = \begin{cases} 1 & \text{if } prefix(w_i) = s \text{ and } y_i = a \\ 0 & otherwise \end{cases}$$

$$\mathbf{f}_{4,w,a,b}(x_{1:n}, i, y_{i-1}, y_i) = \begin{cases} 1 & \text{if } w_i = w \text{ and } y_{i-1} = a \text{ and } y_i = b \\ 0 & otherwise \end{cases}$$

$$\mathbf{f}_{5,w,v,a}(x_{1:n}, i, y_{i-1}, y_i) = \begin{cases} 1 & \text{if } w_{i-2} = w \text{ and } w_{i-1} = v \text{ and } y_i = a \\ 0 & otherwise \end{cases}$$

The formalization of $\mathbf{f}_{6,a,b,c}$ would be:

$$\mathbf{f}_{6,a,b,c}(x_{1:n}, i, y_{i-1}, y_i) = \begin{cases} 1 & \text{if } y_{i-2} = a \text{ and } y_{i-1} = b \text{ and } y_i = c \\ 0 & otherwise \end{cases}$$

However, since y_{i-2} is not part of the considered context in a bigram-factored model, this feature makes no sense.

Exercise 6.

1. Given the training example `the/DT dog/NN saw/VBD the/DT man/NN`, if we convert it to (x, y) pairs for training a trigram-factored log-linear model for PoS tagging, which of the following pairs are in the training set?

   (a) $x = (\texttt{DT}, \texttt{NN}, \texttt{the dog saw the man}, 3); y = \texttt{NN}$

   (b) $x = (\texttt{VBD}, \texttt{DT}, \texttt{the dog saw the man}, 3); y = \texttt{VBD}$

   (c) $x = (\texttt{DT}, \texttt{NN}, \texttt{the dog saw the man}, 3); y = \texttt{VBD}$

   (d) $x = (\texttt{DT}, \texttt{NN}, \texttt{the dog saw the man}, 4); y = \texttt{NN}$

2. List all (x, y) pairs that can be generated from this training set. Assume phantom tags $y_{-1} = y_0 = \texttt{START}$.

SOLUTION

1. (a) does not match because position 3 is has $y = \texttt{VBD}$ and not `NN`. Similarly, (d) is discarded because position 4 has $y = \texttt{DT}$ and not `NN`. Pattern (b) is not matched because previous tags y_{i-2}, y_{i-1} for position 3 are `DT` and `NN` respectively, and not `VBD` and `DT`. The only pair matching the trainig set is (c), since the tag for word 3 is $y = \texttt{VBD}$, and the previous tags y_{i-2}, y_{i-1} are `DT` and `NN` respectively.

2. Pairs generated by this training set are:

   $x = (\texttt{START}, \texttt{START}, \texttt{the dog saw the man}, 1); y = \texttt{DT}$

   $x = (\texttt{START}, \texttt{DT}, \texttt{the dog saw the man}, 2); y = \texttt{NN}$

   $x = (\texttt{DT}, \texttt{NN}, \texttt{the dog saw the man}, 3); y = \texttt{VBD}$

   $x = (\texttt{NN}, \texttt{VBD}, \texttt{the dog saw the man}, 4); y = \texttt{DT}$

   $x = (\texttt{VBD}, \texttt{DT}, \texttt{the dog saw the man}, 5); y = \texttt{NN}$

Exercise 7.

We want to approach a PoS tagging task with a bigram-factored log-linear model that will compute the tag for each word as:

$$tag(x_{1:n}, i) = \underset{y_i \in \mathcal{Y}}{\operatorname{argmax}}\, w \cdot f(x_{1:n}, i, y_{i-1}, y_i)$$

We have defined the following feature function types:

- Type 1: Current tag is a:

$$f_{1,a}(x_{1:n}, i, y_{i-1}, y_i) = \begin{cases} 1 & \text{if } y_i = a \\ 0 & otherwise \end{cases}$$

- Type 2: Current word is capitalized and current tag is a:

$$f_{2,a}(x_{1:n}, i, y_{i-1}, y_i) = \begin{cases} 1 & \text{if } x_i \text{ is capitalized and } y_i = a \\ 0 & \text{otherwise} \end{cases}$$

- Type 3: Current tag is a and previous tag is b:

$$f_{3,a,b}(x_{1:n}, i, y_{i-1}, y_i) = \begin{cases} 1 & \text{if } y_{i-1} = a \text{ and } y_i = b \\ 0 & \text{otherwise} \end{cases}$$

1. Propose values for appropriate features in vector w that will correctly classify all words in the following sentences. Try to set the minimum number of non-zero weights. Proof or justification of the chosen values is required.

x:	John	programs	bugs
y:	E	V	N

x:	Mary	runs	programs
y:	E	V	N

x:	Mary	bugs	John
y:	E	V	E

x:	programs	print	results
y:	N	V	N

SOLUTION

- $w_{1,\texttt{V}} = 1$ will score +1 for any word to be tagged as a verb. This will solve correctly all verbs, and introduce a wrong biass in the other words.
- $w_{3,V,N} = 2$ will score +2 in favor of tag `N` for any word after a `V`. This will overcome the first feature and correctly solve the nouns *results*, *bugs*, and *programs*.
- $w_{3,START,N} = 3$ will score +3 in favor of tag `N` for any word at the beggining of the sentence. This will overcome the first feature and correctly solve the noun *programs* in the last sentence. If we used +2 here we would get a tie between combinations V-N and N-V at sentence beggining, so we use +3.
- $w_{2,\texttt{E}} = 4$ will score +4 for all capitalized words to be tagged `E`. This will overcome the previous features and properly solve all occurrences of *John* and *Mary*.

Let's apply Viterbi algorithm to check these weights work for all given sentences:

	John	programs	bugs	
	Mary	runs	programs	
E	E:4	**E-E: 4+0=4**	E-E-E: 4+0=4	
		N-E: 3+0=3	E-N-E: 4+0=4	
		V-E 1+0=1	**E-V-E: 5+0=5**	
N	N:3	**E-N: 4+0=4**	E-E-N: 4+0=0	
		N-N: 3+0=3	E-N-N: 4+0=0	
		V-N: 1+2=3	**E-V-N: 5+2=7**	←Best
V	V: 1	**E-V: 4+1=5**	E-E-V: 4+1=5	
		N-V: 3+1=4	E-N-V: 4+1=5	
		V-V: 1+1=2	**E-V-V: 5+1=6**	

For sentences *John programs bugs* and *Mary runs programs*, the best sequence is E-V-N with a score of 7, higher than any other combination.

	Mary	bugs	John	
E	E: 4	**E-E: 4+0=4**	E-E-E: 4+4=8	
		N-E: 3+0=3	E-N-E: 4+4=8	
		V-E 1+0=1	**E-V-E: 5+4=9**	←Best
N	N: 3	**E-N: 4+0=4**	E-E-N: 4+0=0	
		N-N: 3+0=3	E-N-N: 4+0=0	
		V-N: 1+2=3	**E-V-N: 5+2=7**	
V	V:1	**E-V: 4+1=5**	E-E-V: 4+1=5	
		N-V: 3+1=4	E-N-V: 4+1=5	
		V-V: 1+1=2	**E-V-V: 5+1=6**	

For sentence *Mary bugs John*, the best sequence is E-V-E with a score of 9, higher than any other combination.

	programs	print	results	
E	E: 0	E-E: 0+0=0	N-E-E: 3+0=3	
		N-E: 3+0=3	N-N-E: 3+0=3	
		V-E 1+0=1	**N-V-E: 4+0=4**	
N	N: 3	E-N: 0+0=0	N-E-N: 3+0=3	
		N-N: 3+0=3	N-N-N: 3+0=0	
		V-N: 1+2=3	**N-V-N: 4+2=6**	←Best
V	V: 1	E-V: 0+1=1	N-E-V: 3+1=4	
		N-V: 3+1=4	N-N-V: 3+1=4	
		V-V: 1+1=2	**N-V-V: 4+1=5**	

For sentence *programs print results*, the best sequence is N-V-N with a score of 6, higher than any other combination.

Exercise 8.

We are performing PoS tagging with a trigram-factored CRF, using tagset $\mathcal{T} = \{\texttt{DT}, \texttt{V}, \texttt{NN}, \texttt{ADV}, \texttt{PREP}\}$, and we defined a history as $h = \langle t_{i-2}, t_{i-1}, w_{[1:n]}, i\rangle$.

1. How many possible histories are there for a given input sequence $\mathcal{X}$ and a fixed value of i?
2. Which of the following are valid features?

$$\mathbf{f}_1(h,t) = \begin{cases} 1 & \text{if } t = \texttt{V} \text{ and } t_{i-1} = \texttt{PREP} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{f}_2(h,t) = \begin{cases} 1 & \text{if } t = \texttt{V} \text{ and } w_{i-2} = \texttt{dog} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{f}_3(h,t) = \begin{cases} 1 & \text{if } t = \texttt{V} \text{ and } t_{i-3} = \texttt{NN} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{f}_4(h,t) = \begin{cases} 1 & \text{if } t = \texttt{V} \text{ and } t_{i+1} = \texttt{PREP} \text{ and } w_2 = \texttt{cow} \\ 0 & \text{otherwise} \end{cases}$$

3. Compute the global feature vector $\mathbf{f}(\mathcal{X}, \mathcal{Y})$ for the input sequence is $\mathcal{X}$ =`the dog walked to a park` and the tag sequence $\mathcal{Y}$ = `DT NN V PREP DT NN`, when using the following features:

$$\mathbf{f}_1(h,t) = \begin{cases} 1 & \text{if } t = \texttt{NN} \text{ and } w_i = \texttt{dog} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{f}_2(h,t) = \begin{cases} 1 & \text{if } t = \texttt{NN} \text{ and } t_{i-1} = \texttt{DT} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{f}_3(h,t) = \begin{cases} 1 & \text{if } t = \texttt{NN} \text{ and } t_{i-1} = \texttt{DT} \text{ and } w_{i-1} = \texttt{the} \\ 0 & \text{otherwise} \end{cases}$$

4. Given the history $h = (t_{i-2}, t_{i-1}, w_{[1:n]}, 5) = (\texttt{V}, \texttt{DT}, \texttt{the man saw the dog in the park}, 5)$, which of the following features yield $\mathbf{f}(h, \texttt{NN}) = 1$?

$$\mathbf{f}_1(h,t) = \begin{cases} 1 & \text{if } t = \texttt{NN} \text{ and } w_i = \texttt{dog} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{f}_2(h,t) = \begin{cases} 1 & \text{if } t = \texttt{DT} \text{ and } w_i = \texttt{dog} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{f}_3(h,t) = \begin{cases} 1 & \text{if } t = \texttt{NN} \text{ and } w_{i+1} = \texttt{dog} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{f}_4(h,t) = \begin{cases} 1 & \text{if } t = \texttt{NN} \text{ and } t_{i-1} = \texttt{DT} \\ 0 & \text{otherwise} \end{cases}$$

SOLUTION

1. Each history has the form $h = \langle t_{i-2}, t_{i-1}, w_{[1:n]}, i\rangle$. If we fix $\mathcal{X} = w_{[1:n]}$ and the position i, there are only two parameters left: t_{i-2} and t_{i-1}. Since each of them can take any of the possible five PoS tag values $\{\texttt{DT}, \texttt{V}, \texttt{NN}, \texttt{ADV}, \texttt{PREP}\}$, the number of posible combinations is $5 \times 5 = 25$.
2. Features f_1 and f_2 are valid beacuse they use elements in h (i.e. t_{i-1} and w_{i-2}, respectively). Feature f_3 is invalid because t_{i-3} is not included in h. Feature f_4 is not valid beacuse t_{i+1} is not included in h
3. Given the values of $\mathcal{X}$ and $\mathcal{Y}$, for each word we obtain the following features:

$\mathcal{X}$	the	dog	walked	to	a	park
$\mathcal{Y}$	DT	NN	V	PREP	DT	NN
Features	-	f_1 f_2 f_3	-	-	-	f_2

Thus, the vector resulting of applying given features is: $(f_1, f_2, f_3) = (1, 2, 1)$

4. Position $i = 5$ corresponds to word `dog` in the sentence. Thus, when evaluating each feature for $t = \texttt{NN}$, we get that:

   $f_1(h, \texttt{NN}) = 1$, since $t = \texttt{NN}$ and $w_5 = \texttt{dog}$
   $f_2(h, \texttt{NN}) = 0$, since $t \neq \texttt{DT}$
   $f_3(h, \texttt{NN}) = 0$, since $w_6 \neq \texttt{dog}$
   $f_4(h, \texttt{NN}) = 1$, since $t = \texttt{NN}$ and $t_{i-1} = \texttt{DT}$

Exercise 9.

We want to address a Named Entity Recognition task consisting in identifying diseases in medical texts. For this, we want to train a sequence classifier such as a CRF using bigram factorization (i.e. only previous and current tag hypothesis are considered). Thus, the used context is $h = (t_{i-1}, w_{[1:n]}, pos_{[1:n]}, i)$.

We use the following feature templates:

$$\mathbf{f}_1(h,t) = \begin{cases} 1 & \text{if } pos_{i-1} = \text{N and } t_{i-1} = \text{O} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{f}_2(h,t) = \begin{cases} 1 & \text{if } suf(w_{i-1}) = \text{'ing' and } t_i = \text{B} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{f}_{3,a,b}(h,t) = \begin{cases} 1 & \text{if } w_{i-1} = a \text{ and } t = b \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{f}_{4,a,c}(h,t) = \begin{cases} 1 & \text{if } w_{i-1} = a \text{ and } pos_i = c \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{f}_{5,a}(h,t) = \begin{cases} 1 & \text{if } w_i = a \text{ and } capitalized(w_{i-1}) \\ 0 & \text{otherwise} \end{cases}$$

Given the above templates, and the training sentence:

i	1	2	3	4	5	6	7	8	9	10	11	12	13
w	Fragile-X	syndrome	is	an	inherited	form	of	mental	retardation	involving	mitral	valve	prolapse
pos	N	N	V	D	JJ	N	P	JJ	N	V	JJ	N	N
t	B	I	O	O	O	O	O	O	O	O	B	I	I

List which feature instances would be generated for words:

a) $i = 2$ (*syndrome*)

b) $i = 10$ (*involving*)

c) $i = 11$ (*mitral*)

SOLUTION

a) Features for $i = 2$ (*syndrome*): $(f_{3,\text{Fragile-X,I}}, f_{4,\text{Fragile-X,N}}, f_{5,\text{syndrome}})$

b) Features for $i = 10$ (*involving*): $(f_1, f_{3,\text{retardation,O}}, f_{4,\text{retardation,V}})$

c) Features for $i = 11$ (*mitral*): $(f_2, f_{3,\text{involving,B}}, f_{4,\text{involving,JJ}})$

Exercise 10. Features for log linear sequence annotation models

We are performing PoS tagging for a recently discovered alien language, using a trigram-factored CRF, using tagset $\mathcal{T} = \{\texttt{D}, \texttt{V}, \texttt{N}, \texttt{A}, \texttt{P}\}$, and we defined a history as $h = \langle t_{i-2}, t_{i-1}, w_{[1:n]}, i\rangle$.

1. How many possible histories are there for a given input sequence $\mathcal{X}$ and a fixed value of i? Justify your answer.

2. Which of the following are valid features and which are not? Justify your answer.

$$\mathbf{f}_1(h,t) = \begin{cases} 1 & \text{if } t = \texttt{V} \text{ and } t_{i-1} = \texttt{N} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{f}_2(h,t) = \begin{cases} 1 & \text{if } t = \texttt{K} \text{ and } w_{i-2} = \texttt{skjkeg} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{f}_3(h,t) = \begin{cases} 1 & \text{if } t = \texttt{N} \text{ and } t_{i-3} = \texttt{P} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{f}_4(h,t) = \begin{cases} 1 & \text{if } t = \texttt{V} \text{ and } t_{i+1} = \texttt{A} \text{ and } w_2 = \texttt{wuakla} \\ 0 & \text{otherwise} \end{cases}$$

3. Compute the feature vectors $\mathbf{f}(h,t)$ for each position i, and the global feature vector $\mathbf{f}(\mathcal{X},\mathcal{Y})$ for the input sequence $\mathcal{X}$ = `grufp umdk wuakla du blha skjkeg` and the tag sequence $\mathcal{Y}$ = `P V N D N A`, when using the following features:

$$\mathbf{f}_1(h,t) = \begin{cases} 1 & \text{if } t = \texttt{N} \text{ and } w_i = \texttt{wuakla} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{f}_2(h,t) = \begin{cases} 1 & \text{if } t = \texttt{N} \text{ and } t_{i-1} \neq \texttt{A} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{f}_3(h,t) = \begin{cases} 1 & \text{if } t = \texttt{N} \text{ and } t_{i-1} = \texttt{V} \text{ and } w_{i-1} = \texttt{umdk} \\ 0 & \text{otherwise} \end{cases}$$

SOLUTION

1. For fixed $\mathcal{X}$ and i, the only varible elements of the history are t_{i-2} and t_{i-1}. Since each of them may have any value in $\mathcal{T}$, the number of different possible histories is $|\mathcal{T}|^2 = 5^2 = 25$

2. f_1 is valid, since it depends only on t and t_{i-1}, which is included in h.
   f_2 is not valid, since $\texttt{K} \notin \mathcal{T}$.
   f_3 is not valid, since t_{i-3} is not included in h.
   f_4 is not valid, since t_{i+1} is not included in h.

3. for $i = 1$, we have $h = \langle \texttt{START}, \texttt{START}, \mathcal{X}, 1\rangle$, and $\mathbf{f}(h,t) = (\mathbf{f}_1(h,\texttt{P}), \mathbf{f}_2(h,\texttt{P}), \mathbf{f}_3(h,\texttt{P})) = (0,0,0)$
   for $i = 2$, we have $h = \langle \texttt{START}, \texttt{P}, \mathcal{X}, 2\rangle$, and $\mathbf{f}(h,t) = (\mathbf{f}_1(h,\texttt{V}), \mathbf{f}_2(h,\texttt{V}), \mathbf{f}_3(h,\texttt{V})) = (0,0,0)$
   for $i = 3$, we have $h = \langle \texttt{P}, \texttt{V}, \mathcal{X}, 3\rangle$, and $\mathbf{f}(h,t) = (\mathbf{f}_1(h,\texttt{N}), \mathbf{f}_2(h,\texttt{N}), \mathbf{f}_3(h,\texttt{N})) = (1,1,1)$
   for $i = 4$, we have $h = \langle \texttt{V}, \texttt{N}, \mathcal{X}, 4\rangle$, and $\mathbf{f}(h,t) = (\mathbf{f}_1(h,\texttt{D}), \mathbf{f}_2(h,\texttt{D}), \mathbf{f}_3(h,\texttt{D})) = (0,0,0)$
   for $i = 5$, we have $h = \langle \texttt{N}, \texttt{D}, \mathcal{X}, 5\rangle$, and $\mathbf{f}(h,t) = (\mathbf{f}_1(h,\texttt{N}), \mathbf{f}_2(h,\texttt{N}), \mathbf{f}_3(h,\texttt{N})) = (0,1,0)$
   for $i = 6$, we have $h = \langle \texttt{D}, \texttt{N}, \mathcal{X}, 6\rangle$, and $\mathbf{f}(h,t) = (\mathbf{f}_1(h,\texttt{A}), \mathbf{f}_2(h,\texttt{A}), \mathbf{f}_3(h,\texttt{A})) = (0,0,0)$

   Thus the global feature vector is the sum of the factored vectors: $\mathbf{f}(\mathcal{X},\mathcal{Y}) = (1,2,1)$

Exercise 11.

Negation detection is a task consisting on identifying which phrases in a sentence are affected by a negation. It is a vital task e.g. in applications related to the processing of medical documents.

The task is often modeled as a B-I-O labeling task, and solved using sequence-labeling algorithms such as CRFs.

We have the following training data:

$\mathcal{X}$	The	patient	does	not	show	any	lung	symptoms	.		
$\mathcal{Y}$	O	O	O	B	I	I	I	I	O		
$\mathcal{X}$	Dark	spots	were	observed	in	lung	X-ray	imaging	.		
$\mathcal{Y}$	O	O	O	O	O	O	O	O	O		
$\mathcal{X}$	Exhoglovifin	never	caused	adverse	reactions	and	should	not	be	banned	.
$\mathcal{Y}$	O	B	I	I	I	O	O	B	I	I	O

And the following feature templates:

$$\mathbf{f}_{1,a,l}(\mathcal{X}, i, t) = \begin{cases} 1 & \text{if } w_i = a \wedge t = l \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{f}_{2,l}(\mathcal{X}, i, t) = \begin{cases} 1 & \text{if } w_{i-1} \in \{no, not, never, any\} \wedge t = l \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{f}_{3,l}(\mathcal{X}, i, t) = \begin{cases} 1 & \text{if } punctuation(w_i) \wedge\ t = l \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{f}_{4,l}(\mathcal{X}, i, t) = \begin{cases} 1 & \text{if } w_{i-1} = dark \wedge\ w_i = spots \wedge\ t = l \\ 0 & \text{otherwise} \end{cases}$$

1. Which is the dimension of the feature space instantiated by this dataset? Justify your answer.
2. Given the following test sentence $\mathcal{X}$ and hypothesis tag sequence $\mathcal{Y}$:

$\mathcal{X}$	X-Ray	results	do	not	show	any	dark	spots	.
$\mathcal{Y}$	O	O	O	B	I	I	I	I	O

   compute the feature vectors $\mathbf{f}(\mathcal{X}, i, t)$ for each position i, and the global feature vector $\mathbf{f}(\mathcal{X}, \mathcal{Y})$. Highlight which features in the global vector that are present in the vector space instantiated by the three training sentences above.

SOLUTION

1. Feature f_1 is instantiated for each combination word-label seen in the training data. Sentence 1 contains 9 combinations. Sentence 2 contains 8 new combinations –combination (.,O) is repeated. Sentence 3 contains 9 new combinations –combinations (not,B) and (.,O) are repeated. Total 9+8+9 = 26 feature instances for template f_1.

   Feature f_2 is instantiated for each occurrence of `not`, `never`, or `any` combined with a label. Sentence 1 contains one occurrence (with label `I`), sentence 2 does not contain any, and sentence 3 contains two more occurrences, also with label `I`, so they generate the same feature $f_{2,I}$. Total, 1 feature instances for template f_2.

   Feature f_3 is instantiated for each occurrence of a punctuation sign combined with a label. Each sentence has one occurrence of the combination (.,O), thus only one instance is generated for f_3.

Feature f_4 is instantiated for each occurrence of *dark spots* combined with a label. This only happens once in sentence 2 (with label O), thus only one instance is generated for f_4.

So, the total number of generated features (i.e. our feature space dimension) is $26+1+1+1 = 29$.

2. Feature vectors for each position are:

$$\mathbf{f}(\mathcal{X}, 1, \texttt{O}) = \{f_{1,XRay,O}\}$$
$$\mathbf{f}(\mathcal{X}, 2, \texttt{O}) = \{f_{1,results,O}\}$$
$$\mathbf{f}(\mathcal{X}, 3, \texttt{O}) = \{f_{1,do,O}\}$$
$$\mathbf{f}(\mathcal{X}, 4, \texttt{B}) = \{f_{1,not,B}\}$$
$$\mathbf{f}(\mathcal{X}, 5, \texttt{I}) = \{f_{1,show,I}, f_{2.I}\}$$
$$\mathbf{f}(\mathcal{X}, 6, \texttt{I}) = \{f_{1,any,I}\}$$
$$\mathbf{f}(\mathcal{X}, 7, \texttt{I}) = \{f_{1,dark,I}, f_{2,I}\}$$
$$\mathbf{f}(\mathcal{X}, 8, \texttt{I}) = \{f_{1,spots,I}, f_{4,I}\}$$
$$\mathbf{f}(\mathcal{X}, 9, \texttt{O}) = \{f_{1,.,O}, f_{3,O}\}$$

Thus, the global feature vector $\mathbf{f}(\mathcal{X}, \mathcal{Y}) = \sum_i \mathbf{f}(\mathcal{X}, i, \mathcal{Y}_i)$ is:

feature	value	in training feature space?
$f_{1,XRay,O}$	1	✓
$f_{1,results,I}$	1	× (word *results* is not in training)
$f_{1,do,O}$	1	× (word *do* is not in training)
$f_{1,not,B}$	1	✓
$f_{1,show,I}$	1	✓
$f_{2,I}$	2	✓
$f_{1,any,I}$	1	✓
$f_{1,dark,I}$	1	× ($f_{1,dark,O}$ appears in training, but $f_{1,dark,I}$ does not)
$f_{1,spots,I}$	1	× ($f_{1,spots,O}$ appears in training, but $f_{1,spots,I}$ does not)
$f_{4,I}$	1	× ($f_{4,O}$ appears in training, but $f_{4,I}$ does not)
$f_{1,.,O}$	1	✓
$f_{3,O}$	1	✓

Exercise 12.

Assume we have defined the following feature templates for a bigram-factored model:

- f_1: the current tag is a and the current word is capitalized

$$\mathbf{f}_{1,a}(x_{1:n}, i, y_{i-1}, y_i) = \begin{cases} 1 & \text{if } \text{is\_capitalized}(x_i) \text{ and } y_i = a \\ 0 & \text{otherwise} \end{cases}$$

- f_2: the current tag is a and the current word is *not* capitalized

$$\mathbf{f}_{2,a}(x_{1:n}, i, y_{i-1}, y_i) = \begin{cases} 1 & \text{if not } \text{is\_capitalized}(x_i) \text{ and } y_i = a \\ 0 & \text{otherwise} \end{cases}$$

- f_3: the previous tag is a, the current tag is b and the current word is capitalized

$$\mathbf{f}_{3,a,b}(x_{1:n}, i, y_{i-1}, y_i) = \begin{cases} 1 & \text{if } \text{is\_capitalized}(x_i) \text{ and } y_{i-1} = a \text{ and } y_i = b \\ 0 & \text{otherwise} \end{cases}$$

- f_4: the current tag is a and the current word is w

$$\mathbf{f}_{4,a,w}(x_{1:n}, i, y_{i-1}, y_i) = \begin{cases} 1 & \text{if } x_i = w \text{ and } y_i = a \\ 0 & \text{otherwise} \end{cases}$$

- f_5: the current tag is a and the next word is w

$$\mathbf{f}_{5,a,w}(x_{1:n}, i, y_{i-1}, y_i) = \begin{cases} 1 & \text{if } x_{i+1} = w \text{ and } y_i = a \\ 0 & \text{otherwise} \end{cases}$$

- f_6: the current tag is a and the previous word is w

$$\mathbf{f}_{6,a,w}(x_{1:n}, i, y_{i-1}, y_i) = \begin{cases} 1 & \text{if } x_{i+1} = w \text{ and } y_i = a \\ 0 & \text{otherwise} \end{cases}$$

- f_7: the previous tag is a and the current tag is b

$$\mathbf{f}_{7,a,b}(x_{1:n}, i, y_{i-1}, y_i) = \begin{cases} 1 & \text{if } y_{i-1} = a \text{ and } y_i = b \\ 0 & \text{otherwise} \end{cases}$$

1. **Perceptron updates**

   Consider the following training example:

y	PER	PER	-	-	LOC	LOC
x	Jack	London	went	to	South	Paris

   Say we are running Perceptron, and under our current **w** the prediction given by Equation 1 is the following sequence **z**:

z	PER	LOC	-	-	-	LOC
x	Jack	London	went	to	South	Paris

Write the perceptron update. That is, write a weight vector g corresponding to $\mathbf{g} = \mathbf{f}(x, y) - \mathbf{f}(x, \mathbf{z})$. Write the non-zero values only.

2. **Setting weights**

   Using the feature definitions above, write a weight vector that correctly classifies all the examples below, i.e. the weight vector should predict the correct tag sequences for all the examples. Try to set as few non-zero weights as possible. Justify your answer.

   - PER - -
     Maria is beautiful
   - LOC - -
     Barcelona is beautiful
   - PER - - LOC
     Jack went to London
   - LOC - -
     Paris is nice
   - PER PER - - LOC LOC
     Jack London went to South Paris
   - ORG - - ORG
     Barcelona played against Paris

SOLUTION

1. **Perceptron updates**

   We are asked to compute $\mathbf{g} = \mathbf{f}(x, y) - \mathbf{f}(x, \mathbf{z})$. So, let's compute $\mathbf{f}(x, y)$ and $\mathbf{f}(x, \mathbf{z})$ and substract them.

   $\mathbf{f}(x, y)$ is the feature vector for the given training example. We compute the active features for each position i:

y	PER	PER	-	-	LOC	LOC
x	Jack	London	went	to	South	Paris
Features	$f_{1,\text{PER}}$	$f_{1,\text{PER}}$	$f_{2,\text{-}}$	$f_{2,\text{-}}$	$f_{1,\text{LOC}}$	$f_{1,\text{LOC}}$
	$f_{3,\text{START,PER}}$	$f_{3,\text{PER,PER}}$	$f_{4,\text{-,went}}$	$f_{4,\text{-,to}}$	$f_{3,\text{-,LOC}}$	$f_{3,\text{LOC,LOC}}$
	$f_{4,\text{PER,Jack}}$	$f_{4,\text{PER,London}}$	$f_{5,\text{-,to}}$	$f_{5,\text{-,South}}$	$f_{4,\text{LOC,South}}$	$f_{4,\text{LOC,Paris}}$
	$f_{5,\text{PER,London}}$	$f_{5,\text{PER,went}}$	$f_{6,\text{-,London}}$	$f_{6,\text{-,went}}$	$f_{5,\text{LOC,Paris}}$	$f_{6,\text{LOC,South}}$
	$f_{7,\text{START,PER}}$	$f_{6,\text{PER,Jack}}$	$f_{7,\text{PER,-}}$	$f_{7,\text{-,-}}$	$f_{6,\text{LOC,to}}$	$f_{7,\text{LOC,LOC}}$
		$f_{7,\text{PER,PER}}$			$f_{7,\text{-,LOC}}$	

   $\mathbf{f}(x, \mathbf{z})$ is the feature vector for the same sentence under current weights. We compute the active features for each position i. Since the sentence is the same, the obtained features will differ only in patterns affecting tags of differently labeled words. Changes with respect to previous table are highlighted in red:

y	PER	LOC	-	-	-	LOC
x	Jack	London	went	to	South	Paris
Features	$f_{1,\text{PER}}$	$f_{1,\text{LOC}}$	$f_{2,\text{-}}$	$f_{2,\text{-}}$	$f_{1,\text{-}}$	$f_{1,\text{LOC}}$
	$f_{3,\text{START,PER}}$	$f_{3,\text{PER,LOC}}$	$f_{4,\text{-,went}}$	$f_{4,\text{-,to}}$	$f_{3,\text{-,-}}$	$f_{3,\text{-,LOC}}$
	$f_{4,\text{PER,Jack}}$	$f_{4,\text{LOC,London}}$	$f_{5,\text{-,to}}$	$f_{5,\text{-,South}}$	$f_{4,\text{-,South}}$	$f_{4,\text{LOC,Paris}}$
	$f_{5,\text{PER,London}}$	$f_{5,\text{LOC,went}}$	$f_{6,\text{-,London}}$	$f_{6,\text{-,went}}$	$f_{5,\text{-,Paris}}$	$f_{6,\text{LOC,South}}$
	$f_{7,\text{START,PER}}$	$f_{6,\text{LOC,Jack}}$	$f_{7,\text{LOC,-}}$	$f_{7,\text{-,-}}$	$f_{6,\text{-,to}}$	$f_{7,\text{-,LOC}}$
		$f_{7,\text{PER,LOC}}$			$f_{7,\text{-,-}}$	

   To compute the difference, it is easier if we tabulate both vectors. Remember that the feature vector for the whole sequence is the sum of the binary vectors obtained at each word position:

Feature	$\mathbf{f}(x, y)$	$\mathbf{f}(x, \mathbf{z})$	$\mathbf{g}$
$f_{1,-}$	-	1	-1
$f_{1,LOC}$	2	2	0
$f_{1,PER}$	2	1	1
$f_{2,-}$	2	2	0
$f_{3,-,-}$	-	1	-1
$f_{3,-,LOC}$	1	1	0
$f_{3,LOC,LOC}$	1	-	1
$f_{3,PER,LOC}$	-	1	-1
$f_{3,PER,PER}$	1	-	1
$f_{3,START,PER}$	1	1	0
$f_{4,LOC,London}$	-	1	-1
$f_{4,LOC,Paris}$	1	1	0
$f_{4,LOC,South}$	1	-	1
$f_{4,PER,Jack}$	1	1	0
$f_{4,PER,London}$	1	-	1
$f_{4,-,South}$	-	1	-1
$f_{4,-,to}$	1	1	0
$f_{4,-,went}$	1	1	0
$f_{5,LOC,Paris}$	1	-	1
$f_{5,LOC,went}$	-	1	-1
$f_{5,-,Paris}$	-	1	-1
$f_{5,PER,London}$	1	1	0
$f_{5,PER,went}$	1	-	1
$f_{5,-,South}$	1	1	0
$f_{5,-,to}$	1	1	0
$f_{6,LOC,Jack}$	-	1	-1
$f_{6,LOC,South}$	1	1	0
$f_{6,LOC,to}$	1	-	1
$f_{6,-,London}$	1	1	0
$f_{6,PER,Jack}$	1	-	1
$f_{6,-,to}$	-	1	-1
$f_{6,-,went}$	1	1	0
$f_{7,-,-}$	1	2	-1
$f_{7,LOC,-}$	-	1	-1
$f_{7,-,LOC}$	1	1	0
$f_{7,LOC,LOC}$	1	-	1
$f_{7,PER,LOC}$	-	1	-1
$f_{7,PER,-}$	1	-	1
$f_{7,PER,PER}$	1	-	1
$f_{7,START,PER}$	-	1	-1

Exercise 13.

Recall the factored linear models of the previous exercises:

$$f(\mathbf{x}_{1:n}) = \operatorname*{argmax}_{\mathbf{y}_{1:n} \in \mathcal{Y}^n} \sum_{i=1}^{n} \mathbf{w} \cdot \mathbf{f}(\mathbf{x}, i, \mathbf{y}_{i-1}, \mathbf{y}_i)$$

In order to compute $f(\mathbf{x}_{1:n})$ we can use the Viterbi algorithm as follows:

- *Define $\delta_i(a)$ to be the score of optimal sequence for $\mathbf{x}_{1:i}$ ending with $a \in \mathcal{Y}$:*

$$\delta_i(a) = \max_{\mathbf{y}_{1:i} \in \mathcal{Y}^i : \mathbf{y}_i = a} \sum_{j=1}^{i} \mathbf{w} \cdot \mathbf{f}(\mathbf{x}, j, \mathbf{y}_{j-1}, \mathbf{y}_j)$$

- *Use the following recursions, for all $a \in \mathcal{Y}$:*

$$\begin{aligned}
\delta_1(a) &= \mathbf{w} \cdot \mathbf{f}(\mathbf{x}, 1, \text{START}, a) \quad \textit{(initialization, assuming } y_0 = \text{START}\textit{)} \\
\delta_i(a) &= \max_{b \in \mathcal{Y}} \delta_{i-1}(b) + \mathbf{w} \cdot \mathbf{f}(\mathbf{x}, i, b, a) \quad \textit{(recursion, } \forall i > 1\textit{)} \\
\gamma_i(a) &= \operatorname*{argmax}_{b \in \mathcal{Y}} \delta_{i-1}(b) + \mathbf{w} \cdot \mathbf{f}(\mathbf{x}, i, b, a) \quad \textit{(backtrace, } \forall i > 1\textit{)}
\end{aligned}$$

- *The optimal score for $\mathbf{x}$ is $\max_{a \in \mathcal{Y}} \delta_n(a)$*
- *The optimal sequence $\hat{\mathbf{y}}$ can be recovered through the backpointers γ*

1. **Viterbi for Trigram Models**

   Assume our model is now factored on trigrams

$$f(\mathbf{x}_{1:n}) = \operatorname*{argmax}_{\mathbf{y}_{1:n} \in \mathcal{Y}^n} \sum_{i=1}^{n} \mathbf{w} \cdot \mathbf{f}(\mathbf{x}, i, \mathbf{y}_{i-2}, \mathbf{y}_{i-1}, \mathbf{y}_i)$$

   That is, $\mathbf{f}(\mathbf{x}, i, \mathbf{y}_{i-2}, \mathbf{y}_{i-1}, \mathbf{y}_i)$ now returns a feature vector for a trigram of the output sequence *ending* at position i. You can assume that $\mathbf{y}_0$ and $\mathbf{y}_{-1}$ are special tags START. Redefine the δ and γ variables and the recursions to compute them.

2. **Viterbi for Trigram Models, Variation**

   Assume our model is now factored on trigrams, but with a slight change in the definition:

$$f(\mathbf{x}_{1:n}) = \operatorname*{argmax}_{\mathbf{y}_{1:n} \in \mathcal{Y}^n} \sum_{i=1}^{n} \mathbf{w} \cdot \mathbf{f}(\mathbf{x}, i, \mathbf{y}_{i-1}, \mathbf{y}_i, \mathbf{y}_{i+1})$$

   Now $\mathbf{f}(\mathbf{x}, i, \mathbf{y}_{i-1}, \mathbf{y}_i, \mathbf{y}_{i+1})$ returns a feature vector for a trigram of the output sequence *centered* at position i. You can assume that $\mathbf{y}_0$ is a special tag START and that $\mathbf{y}_{n+1}$ is a special END tag. Redefine the δ and γ variables and the recursions to compute them.

SOLUTION

1. In viterbi for trigram-factored models we define $\delta_i(b, a)$ to be the score of optimal sequence for $\mathbf{x}_{1:i}$ ending with $y_{i-1} = b, y_i = a$.

$$\begin{aligned}
\delta_1(START, a) &= \mathbf{w} \cdot \mathbf{f}(\mathbf{x}, 1, \text{START}, \text{START}, a) \quad \text{(initialization, } y_{-1} = y_0 = \text{START)} \\
\delta_i(b, a) &= \max_{c \in \mathcal{Y}_{i-2}} \delta_{i-1}(c, b) + \mathbf{w} \cdot \mathbf{f}(\mathbf{x}, i, c, b, a) \quad \text{(recursion, } \forall i > 1) \\
\gamma_i(b, a) &= \operatorname*{argmax}_{c \in \mathcal{Y}_{i-2}} \delta_{i-1}(c, b) + \mathbf{w} \cdot \mathbf{f}(\mathbf{x}, i, c, b, a) \quad \text{(backtrace, } \forall i > 1)
\end{aligned}$$

   where $\mathcal{Y}_i = \{\text{START}\}$ for $i \leq 2$, and $\mathcal{Y}_i = \mathcal{Y}$ for $i > 2$. The optimal score is $\max_{(b,a) \in \mathcal{Y}^2} \delta_n(b, a)$. The optimal sequence is recovered via γ backtraces.

2. In this variation, $\delta_i(b, a)$ computers be the score of optimal sequence for $\mathbf{x}_{1:i}$ ending with $y_i = b, y_{i+1} = a$. It is basically the same solution than above, but shifting table contents one column to the left so we store in δ_i the computations that were stored in δ_{i+1} in previous exercise.

$$\begin{aligned}
\delta_1(b, a) &= \mathbf{w} \cdot \mathbf{f}(\mathbf{x}, 1, \text{START}, b, a) \quad \text{(initialization, } y_0 = \text{START)} \\
\delta_i(b, a) &= \max_{c \in \mathcal{Y}_{i-1}} \delta_{i-1}(c, b) + \mathbf{w} \cdot \mathbf{f}(\mathbf{x}, i, c, b, a) \quad \text{(recursion, } \forall i > 1) \\
\gamma_i(b, a) &= \operatorname*{argmax}_{c \in \mathcal{Y}_{i-1}} \delta_{i-1}(c, b) + \mathbf{w} \cdot \mathbf{f}(\mathbf{x}, i, c, b, a) \quad \text{(backtrace, } \forall i > 1)
\end{aligned}$$

where $\mathcal{Y}_1 = \{\text{START}\}$, $\mathcal{Y}_{n+1} = \{\text{END}\}$, and $\mathcal{Y}_i = \mathcal{Y}$ for $1 < i \leq n$. The optimal score is $\max_{a \in \mathcal{Y}} \delta_n(a, \text{END})$. The optimal sequence is recovered via γ backtraces.