Wie hoch ist die **Entropie** einer Zufallsvariablen mit einer uniformen Verteilung über 4 Werte (also wenn alle Werte die gleiche Wahrscheinlichkeit haben)?

Die **Entropie** misst, wieviel Information ein Zufallsereignis im Mittel enthält.

Entropie einer Zufallsvariablen X mit der Verteilungsfunktion $p(x)$:

$$H(X) = H(p) = -\sum_{x \in X} p(x) \log_2 p(x) = E(log_2 \frac{1}{p(x)})$$

Antwort
Bei einer uniformen Verteilung über 4 Werte hat jeder Wert die Wahrscheinlichkeit 1/4.
Die Entropie beträgt daher - 4 * 0.25 * log_2(1/4) = log_2(4) = 2 Bit

Berechnung
Die Zufallsvariable X hat 4 Werte, jeweils mit p(x) = 1/4
H(X) = - [1/4 * log_2 (1/4) + 1/4 * log_2 (1/4) + 1/4 * log_2 (1/4) + 1/4 * log_2 (1/4)]
= - [4 * 1/4 * log_2 (1/4)]
= - 1 * log_2 (1/4)
= - log_2 (1/4)
= log_2 (1 / (1/4))
= log_2 (4)

log (1/a) = -log a

9. **log (1/a) = -log a** means that the logarithm of 1 divided by some number is equal to the negative logarithm of that number. (This is the exactly the opposite of the rule governing exponents where a number raised to a negative number is equal to 1 divided by that number raised to that power.)

For example:

log (1/2) = - log 2 = -0.301

log (1/3) = - log 3 = -0.477

ln (1/2) = -ln 2 = -0.693

ln (1/3) = -ln 3 = -1.099

ref: http://www.mclph.umn.edu/mathrefresh/logs5.html#:~:text=9.,number%20raised%20to%20that%20power.)

Wie hoch ist die Entropie (einer Zufallsvariablen mit einer uniformen Verteilung über 4 Werte), wenn einer der Werte die Wahrscheinlichkeit 1 besitzt?

$$H(X) = H(p) = -\sum_{x \in X} p(x) \log_2 p(x)$$

Antwort
0
Berechnung:
Wenn eine p(x) den Wert 1 hat, dann sind die anderen p(x) null. Wenn man die Werte in der Formel von Entropie einsetzt, dann bekommt man null als Ergebnis.
H(X) = - [1 * log_2 (1) + 0 * log_2 (0) + 0 * log_2 (0) + 0 * log_2 (0)]

1 * log_2 (1) = 0 (wegen log_2 (1) = 0)

Bei 0 * log_2 (0) wird es wie folgt berechnet.

log(0) ist minus unendlich. Die Frage ist also, welches Resultat 0 mal unendlich ergibt. Hierzu muss der Limes für x->0 von -x log(x) berechnet werden. Da x schneller gegen 0 geht, als -log(x) gegen unendlich geht, ist das Ergebnis 0. Beispiele
x * -log(x)
1/2 * 1 = 1/2
1/4 * 2 = 1/2
1/8 * 3   = 3/8
1/64 * 6 = 1/6
1/1024 * 10 = 1/102.4

Geben Sie die Formel für die Berechnung des Erwartungswertes der Funktion − log p(x) für die Zufallsvariable X an.

Antwort: Die Erwartungswert von −log p(x) = $-\sum_{x \in X} p(x) \log_2 p(x)$ # es ist gleich die Entropie von X

Berechnung:

Erwartungswert einer Funktion f:

$$E(f(X)) = \sum_x p(x) f(x)$$

E(f (X)) ist die Summe über x von p(x) * f(x), wobei x die einzelnen Werte von X sind.
f(X) ist eine Funktion, die eine Zufallsvariable X als Argument nimmt (und X kann mehrere Werte x annimt).
Wir betrachten -log p(x) als eine Funktion, die einen Wert x der Zufallsvariable X als Argument nimmt.
Also, f(x) in der Formel ist hier -log p(x)

p(x) ist die WK der einzelnen Werte x aus X

E(-log p(x)) = SUM_x p(x) * - log p(x) # diese Zeile ist als Antwort auch gültig
= SUM_x p(x) * (-1) * log p(x)
= (-1) SUM_x p(x) * log p(x) # -1 aus der Summe nehmen
= - SUM_x p(x) * log p(x) # so erhalten wir die Entropie-Formel von X

$$\sum_{k=1}^{n} ck = c \sum_{k=1}^{n} k \,,$$

Aufgabe 4) Angenommen der Satz “I can can a can” soll mit einem Bigramm-Tagger und dem Viterbi-Algorithmus mit Wortarten annotiert werden und das Tagset umfasst nur die Tags PRO, D, N, und V. Bei jedem Wort soll jedes Tag erlaubt sein.

1 Wie wird die Tabelle mit den Viterbiwahrscheinlichkeiten zu Beginn initialisiert?

2 Wie berechnet der Algorithmus die Viterbi-Wahrscheinlichkeit des Tags D an Position 4, also beim Wort “a”? (Geben Sie den richtigen Ausdruck für diesen konkreten Fall ohne Variablen an.)

Viterbi-Algorithmus (Bigramm-Tagger)

1. Initialisierung: $\delta_t(0) = \begin{cases} 1 & \text{falls } t = \langle s \rangle \\ 0 & \text{sonst} \end{cases}$

2. Berechnung: (für $0 < k \leq n+1$)

$$\delta_t(k) = \max_{t'} \delta_{t'}(k-1)\, p(t|t')\, p(w_k|t)$$

			k-1	k		
“ “	I	can	can	a	can	“ “
<s>	PRO	PRO	PRO	PRO	PRO	<s>
	D	D	D	D t	D	
	N	N	N t’	N	N	
	V	V	V	V	V	

1. initialisierung: vit(<s>, 0) = 1
2. vit(D,4) = max [vit(PRO,3) * p(D | PRO) * p(a | D),
vit(D,3) * p(D | D) * p(a | D),
vit(N,3) * p(D | N) * p(a | D),
vit(V,3) * p(D | V) * p(a | D)]

Aufgabe 4) Angenommen der Satz "I can can a can" soll mit einem ~~Bigramm-Tagger~~ Trigramm-Tagger und dem Viterbi-Algorithmus mit Wortarten annotiert werden und das Tagset umfasst nur die Tags PRO, D, N, und V. Bei jedem Wort soll jedes Tag erlaubt sein.

1 Wie wird die Tabelle mit den Viterbiwahrscheinlichkeiten zu Beginn initialisiert?

2 Wie berechnet der Algorithmus die Viterbi-Wahrscheinlichkeit des Tags ~~D~~ PRO,D an Position 4, also beim Wort " " (Geben Sie den richtigen Ausdruck für diesen konkreten Fall ohne Variablen an.) (3 Punkte)

				k-1	k			k= n+2
" "	" "	I	can	can	a	can	" "	" "
<s>	<s>	PRO	PRO	PRO t'	PRO	PRO	<s>	<s>
		D	D	D	D t	D		
		N	N t''	N	N	N		
		V	V	V	V	V		

1. Initialisierung: $\delta_{t',t}(0) = \begin{cases} 1 & \text{falls } t = \langle s \rangle \text{ und } t' = \langle s \rangle \\ 0 & \text{sonst} \end{cases}$

2. Berechnung: (für $0 < k \leq n+2$)

$$\delta_{t',t}(k) = \max_{t''} \delta_{t'',t'}(k-1)\, p(t|t'',t')\, p(w_k|t)$$

1. initialisierung: vit(<s>,<s>, 0) = 1
2. vit(PRO,D,4) = max [vit(PRO, PRO,3) * p(D | PRO, PRO) * p(a | D),
vit(D, PRO,3) * p(D | D, PRO) * p(a | D),
vit(N, PRO,3) * p(D | N, PRO) * p(a | D),
vit(V, PRO,3) * p(D | V, PRO) * p(a | D)]

Aufgabe 4) Angenommen der Satz "I can can a can" soll mit einem ~~Bigramm-Tagger~~ Trigramm-Tagger und dem Viterbi-Algorithmus mit Wortarten annotiert werden und das Tagset umfasst nur die Tags PRO, D, N, und V. Bei jedem Wort soll jedes Tag erlaubt sein.

1 Wie wird die Tabelle mit den Viterbiwahrscheinlichkeiten zu Beginn initialisiert?

2 Wie berechnet der Algorithmus die Viterbi-Wahrscheinlichkeit des Tags ~~D~~ <s>,<s> an Position k=n+2 beim Wort "a"? (Geben Sie den richtigen Ausdruck für diesen konkreten Fall ohne Variablen an.) (3 Punkte)

							k-1	k = n+2
" "	" "	I	can	can	a	can	" "	" "
<s>	<s>	PRO	PRO	PRO	PRO	PRO	<s>	<s>
		D	D	D	D	D		
		N	N	N	N	N	t'	t
		V	V	V	V	V		
						t''		

1. Initialisierung: $\delta_{t',t}(0) = \begin{cases} 1 & \text{falls } t = \langle s \rangle \text{ und } t' = \langle s \rangle \\ 0 & \text{sonst} \end{cases}$

2. Berechnung: (für $0 < k \leq n+2$)

$$\delta_{t',t}(k) = \max_{t''} \delta_{t'',t'}(k-1)\, p(t|t'', t')\, p(w_k|t)$$

1. initialisierung: vit(<s>,<s>, 0) = 1
2. vit(<s>,<s>,7) = max [vit(PRO, <s>,6) * p(<s> | PRO, <s>) * p(" " | <s>),
vit(D, <s>,6) * p(<s> | D, <s>) * p(" " | <s>),
vit(N, <s>,6) * p(<s> | N, <s>) * p(" " | <s>),
vit(V, <s>,6) * p(<s> | V, <s>) * p(" " | <s>)]

CRF Viterbi

Merkmale:

- Tag-Wort-Merkmale   Beispiel: TW-INJ-Hallo mit Gewicht 0.7
- Tag-Tag-Merkmale   Beispiel: TT-$\langle s \rangle$-INJ   mit Gewicht 1.3

Berechnung eines **Merkmalsvektors**

f($\langle s \rangle$, INJ, (Hallo),1) = {("TT-$\langle s \rangle$-INJ", 1), ("TW-INJ-Hallo", 1)}

Berechnung eines **lokalen Scores**

s($\langle s \rangle$, INJ, (Hallo),1) = 1.3 * 1 + 0.7 * 1 = 2.0

Berechnung der **Viterbi-Scores** $\delta(i, t)$:

$$
\begin{aligned}
\delta(0, \langle s \rangle) &= log(1) = 0 \\
\delta(1, INJ) &= max_{t \in \{\langle s \rangle\}}\ \delta(0, t) + s(t, INJ, (Hallo), 1) = 0 + 2.0 \\
\delta(1, XY) &= max_{t \in \{\langle s \rangle\}}\ \delta(0, t) + s(t, XY, (Hallo), 1) = 0 + ... \\
\delta(2, \langle s \rangle) &= max_{t \in \{INJ, XY\}}\ \delta(1, t) + s(t, \langle s \rangle, (Hallo), 2) = ...
\end{aligned}
$$

Übung: Berechne vit(D,4) für Wort a (Viterbi, Bigramm, CRF)

Alternativ mit logarithmierten Werten:

$$
\begin{aligned}
s(t', t, w_1^n, i) &= \theta \cdot f(t', t, w_1^n, i) \\
\delta_t(0) &= 0 \text{ falls } t = \langle s \rangle \text{ sonst } -\inf \\
\delta_t(i) &= \max_{t'} \delta_{t'}(i-1) + s(t', t, w_1^n, i) \quad \text{für } 1 \le i \le n+1 \\
\psi_t(i) &= \arg\max_{t'} \delta_{t'}(i-1) + s(t', t, w_1^n, i) \quad \text{für } 1 \le i \le n+1
\end{aligned}
$$

			i-1	i		
" "	I	can	can	a	can	" "
<s>	PRO	PRO	PRO	PRO	PRO	<s>
	D	D	D	D t	D	
	N	N	N t'	N	N	
	V	V	V	V	V	

1. initialisierung: vit(<s>, 0) = 0
2. vit(D,4) = max [vit(PRO,3) + s(PRO, D, "I can can a can", 4),
   vit(D,3) + s(D, D, "I can can a can", 4),
   vit(N,3) + s(N, D, "I can can a can", 4),
   vit(V,3) + s(V, D, "I can can a can", 4)]

Falls Merkmal = tag-tag, tag-wort (wie bei HMM), haben wir f (PRO, D, "I can can a can", 4) = {"TT-PRO-D": 1, "TW-D-a": 1 }

s(PRO, D, "I can can a can", 4) = gewicht("TT-PRO-D") * 1 + gewicht("TW-D-a") *1
falls gewichte(PRO, D, "I can can a can", 4) = {"TT-PRO-D": 0.3, "TW-D-a": 2.4 }, dann haben wir
s(PRO, D, "I can can a can", 4) = 0.3 * 1 + 2.4 *1 oder einfach 0.3 + 2.4

Übung: Berechne **vit(PRO, D,4)** (Viterbi, **Trigramm**, CRF)

1. Initialisierung: $\delta_{t',t}(0) = \begin{cases} 1 & \text{falls } t = \langle s \rangle \text{ und } t' = \langle s \rangle \\ 0 & \text{sonst} \end{cases}$

2. Berechnung: (für $0 < k \leq n+2$)

+ s(t'', t', t, w1..n, i)

$$\delta_{t',t}(k) \;=\; \max_{t''} \delta_{t'',t'}(k-1)\ \cancel{p(t|t'',t')\ p(w_k|t)}$$

note: hier ist die Bigramm Formel von CRF-Viterbi (steht nur als Referenz hier)

$$\begin{aligned} s(t',t,w_1^n,i) &= \theta \cdot f(t',t,w_1^n,i) \\ \delta_t(0) &= 0 \text{ falls } t = \langle s \rangle \text{ sonst } -\inf \\ \delta_t(i) &= \max_{t'} \delta_{t'}(i-1) + s(t',t,w_1^n,i) \quad \text{für } 1 \leq i \leq n+1 \end{aligned}$$

					k-1		k				
" "	" "	I	can		can		a		can	" "	" "
<s>	<s>	PRO	PRO		PRO	t'	PRO		PRO	<s>	<s>
		D	D		D		D	t	D		
		N	N	t''	N		N		N		
		V	V		V		V		V		

1. initialisierung: vit(<s>, 0) = 0
2. vit(PRO,D,4) = max [vit(PRO,PRO,3) + s(PRO, PRO, D, "I can can a can", 4),
   vit(D,PRO,3) + s(D, PRO, D, "I can can a can", 4),
   vit(N,PRO,3) + s(N, PRO, D, "I can can a can", 4),
   vit(V,PRO,3) + s(V, PRO, D, "I can can a can", 4)]

Merkmale kann z.B. wie folg definieren: "TTT- T1-T2-T3", "TW-T3-W3" , mit T3 entspricht t , T2 entspricht t', und T1 entspricht t''
Beispiel:
f(PRO, PRO, D, "I can can a can", 4) = { "TTT-PRO-PRO-D": 1, "TW-D-a": 1 }

Andere mögliche Merkmale

- Für jede Wortposition i wird ein Merkmalsvektor $f(t_{i-k}^{i}, w_1^n, i)$ extrahiert.
  - lexikalische Merkmale (ohne Nachbartag-Information)
    Wort+Tag, Wortsuffix+Tag, Wort"shape"+Tag, voriges Wort+Tag, nächstes Wort+Wort+Tag, Wort+Tag im Lexikon
  - Kontext-Merkmale (mit Nachbartag-Information)
    voriges Tag+Tag, letzte2Tags+Tag, voriges Tag+Wort+Tag

Übung: Trigramm-Forward-Wk
1. initialisiere die Forward WK nach der Formel
2. Berechne Forward-WK für PRO,D an Position 4 für die Wortfolge “I can can a can”. Alle Wörter haben die gleichen möglichen Tags: PRO,D, N, V.

Definition der Forward-Wahrscheinlichkeiten $\alpha_{t',t}(k)$

$$\alpha_{t',t}(0) = \begin{cases} 1 & \text{falls } t = t' = \langle s \rangle \\ 0 & \text{sonst} \end{cases}$$

$$\alpha_{t',t}(k) = \sum_{t'' \in T} \alpha_{t'',t'}(k-1)\, p(t|t'',t')\, p(w_k|t) \qquad \text{für } 1 \leq k \leq \mathbf{n+2}$$

I	can	can	a	can
PRO	PRO	PRO	PRO	PRO
D	D	D	D	D
N	N	N	N	N
V	V	V	V	V

Übung (Lösung):
1. initialisierung
2. Berechne Forward-WK für PRO,D an Position 4 für die Wortfolge “I can can a can”. Alle Wörter haben die gleichen möglichen Tags: PRO,D, N, V.

Definition der Forward-Wahrscheinlichkeiten $\alpha_{t',t}(k)$

$$
\begin{aligned}
\alpha_{t',t}(0) &= \begin{cases} 1 & \text{falls } t = t' = \langle s \rangle \\ 0 & \text{sonst} \end{cases} \\
\alpha_{t',t}(k) &= \sum_{t'' \in T} \alpha_{t'',t'}(k-1)\, p(t|t'',t')\, p(w_k|t) \qquad \text{für } 1 \le k \le \mathbf{n+2}
\end{aligned}
$$

“ “	“ “	I	can	can	a	can	“ “	“ “
<s>	<s>	PRO	PRO	PRO	PRO	PRO	<s>	<s>
		D	D	D	D	D		
		N	N	N	N	N		
		V	V	V	V	V		

1. fw(<s>,<s>,0) = 1
2. fw(PRO,D,4) = fw(PRO,PRO,3) * p(D| PRO,PRO) * p(a | D) +
fw(D,PRO,3) * p(D| D,PRO) * p(a | D) +
fw(N,PRO,3) * p(D| N,PRO) * p(a | D) +
fw(V,PRO,3) * p(D| V,PRO) * p(a | D)

Übung: Trigramm-Backward-Wk

Definition der Backward-Wahrscheinlichkeiten $\beta_{t'',t'}(k)$:

$$\beta_{t'',t'}(n+2) = \begin{cases} 1 & \text{falls } t'' = t' = \langle s \rangle \\ 0 & \text{sonst} \end{cases}$$

$$\beta_{t'',t'}(k) = \sum_{t \in T} p(t|t'',t')\, p(w_{k+1}|t)\, \beta_{t',t}(k+1) \qquad \text{für } 0 \leq k \leq n+1$$

“ “	“ “	I	can	can	a	can	“ “	“ “
<s>	<s>	PRO	PRO	PRO	PRO	PRO	<s>	<s>
		D	D	D	D	D		
		N	N	N	N	N		
		V	V	V	V	V		

1. initialisierung
2. Bereche die Backward Wk für PRO,D an position 4

Übung (Lösung): Trigramm-Backward Wk

Definition der Backward-Wahrscheinlichkeiten $\beta_{t'',t'}(k)$:

$$\beta_{t'',t'}(n+2) = \begin{cases} 1 & \text{falls } t'' = t' = \langle s \rangle \\ 0 & \text{sonst} \end{cases}$$

$$\beta_{t'',t'}(k) = \sum_{t \in T} p(t|t'',t')\, p(w_{k+1}|t)\, \beta_{t',t}(k+1) \qquad \text{für } 0 \leq k \leq n+1$$

					k	k +1		
“ “	“ “	I	can	can	a	can	“ “	“ “
				t”	t’	t		
<s>	<s>	PRO	PRO	PRO	PRO	PRO	<s>	<s>
		D	D	D	D	D		
		N	N	N	N	N		
		V	V	V	V	V		

1. initialisierung
   bw(<s>,<s>,7) = 1
2. Bereche die Backward Wk für PRO,D an position 4
   bw(PRO,D,4) = p(PRO | PRO, D) * p(can | PRO) * bw(D,PRO, 5) +
   p(D | PRO, D) * p(can | D) * bw(D,D, 5) +
   p(N | PRO, D) * p(can | N) * bw(D,N, 5) +
   p(V | PRO, D) * p(can | V) * bw(D,V, 5)

Übung: Bigramm-Forward Wk

$$\alpha_t(0) = \begin{cases} 1 & \text{falls } t = \langle s \rangle \\ 0 & \text{sonst} \end{cases}$$

$$\alpha_t(k) = \sum_{t' \in T} \alpha_{t'}(k-1)\, p(t|t')\, p(w_k|t) \quad \text{für } 0 < k \leq n+1$$

“ “	I	can	can	a	can	“ “
<s>	PRO	PRO	PRO	PRO	PRO	<s>
	D	D	D	D	D	
	N	N	N	N	N	
	V	V	V	V	V	

1. initialisierung
2. Bereche die Forward -Wk für D an position 4

Übung (Lösung) : Bigramm-Forward Wk

$$\alpha_t(0) = \begin{cases} 1 & \text{falls } t = \langle s \rangle \\ 0 & \text{sonst} \end{cases}$$

$$\alpha_t(k) = \sum_{t' \in T} \alpha_{t'}(k-1)\, p(t|t')\, p(w_k|t) \qquad \text{für } 0 < k \leq n+1$$

			k-1	k		
" "	I	can	can	a	can	" "
<s>	PRO	PRO	PRO	PRO	PRO	<s>
	D	D	D	D t	D	
	N	N	N	N	N	
	V	V	V	V	V	
			t'			

1. initialisierung
   fw(<s>, 0) = 1
2. Bereche die Forward -Wk für D an position 4
   fw(D,4) = fw(PRO,3) * p(D | PRO) * p(a | D) +
   fw(D,3) * p(D | D) * p(a | D) +
   fw(N,3) * p(D | N) * p(a | D) +
   fw(V,3) * p(D | V) * p(a | D)

Übung: Bigram-Backward Wk

$$\beta_t(n+1) = \begin{cases} 1 & \text{falls } t = \langle s \rangle \\ 0 & \text{sonst} \end{cases}$$

$$\beta_t(k-1) = \sum_{t' \in T} p(t'|t)\, p(w_k|t')\, \beta_{t'}(k) \qquad \text{für } 0 < k \leq n+1$$

“ “	I	can	can	a	can	“ “
<s>	PRO	PRO	PRO	PRO	PRO	<s>
	D	D	D	D	D	
	N	N	N	N	N	
	V	V	V	V	V	

1. initialisierung
2. Bereche die Backward -Wk für D an position 4

Übung(Lösung): Bigram-Backward Wk

$$\beta_t(n+1) = \begin{cases} 1 & \text{falls } t = \langle s \rangle \\ 0 & \text{sonst} \end{cases}$$

$$\beta_t(k-1) = \sum_{t' \in T} p(t'|t)\, p(w_k|t')\, \beta_{t'}(k) \quad \text{für } 0 < k \leq n+1$$

				k-1	k	
“ “	I	can	can	a	can	“ “
<s>	PRO	PRO	PRO	PRO	PRO	<s>
	D	D	D	D t	D	
	N	N	N	N	N	
	V	V	V	V	V	
					t’	

1. initialisierung
   bw(<s>,0) = 1
2. Bereche die Backward -Wk für D an position 4
   bw(D,4) = p (PRO | D) * p(can | PRO) * bw(PRO, 5) +
   p (D | D) * p(can | D) * bw(D, 5) +
   p (N | D) * p(can | N) * bw(N, 5) +
   p (V | D) * p(can | V) * bw(V, 5)

Aufgabe 6) Wie ist beim Conditional Random Field die Wahrscheinlichkeit einer Tagfolge definiert? (4 Punkte)

$$p(t_1^n|w_1^n) = \frac{1}{Z(w_1^n)} e^{\theta \cdot \sum_{i=1}^{n+1} f(t_{i-k}^i, w_1^n, i)}$$